# Nouvelles méthodes de calcul pour la prédiction des interactions protéine-protéine au niveau structural

Petr Popov

▶ **To cite this version:**

Petr Popov. Nouvelles méthodes de calcul pour la prédiction des interactions protéine-protéine au niveau structural. Mathématiques générales [math.GM]. Université Grenoble Alpes, 2015. Français. NNT : 2015GREAM005 . tel-01167112

**HAL Id: tel-01167112**
**https://theses.hal.science/tel-01167112**

Submitted on 23 Jun 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

# THÈSE

Pour obtenir le grade de

## DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité Mathématiques-Informatique

Arrêté ministériel : 7 août 2006

Présentée par


## Petr POPOV

Directeur de Thèse **Sergei GRUDININ**
Co-Directeur de Thèse **Anatoli IOUDITSKI**
Co-Encadrant de Thèse **Stephane REDON**

préparée au sein du **Laboratoire Nano-D, Inria**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**


# Nouvelles méthodes de calcul pour la prédiction des interactions protéine-protéine au niveau structural

Thèse soutenue publiquement le **« 28 Janvier 2015 »**,
devant le jury composé de :

**Dr. Sergei GRUDININ** **Directeur de Thèse**
Chargé de recherche, CNRS, Grenoble, France
**Prof. Dr. Anatoli IOUDITSKI** **Co-Directeur de Thèse**
Professeur UJF, LJK, Grenoble, France
**Dr. Stephane REDON** **Co-Encadrant de Thèse**
Chargé de recherche, Inria, Grenoble, France
**Prof. Dr. Frederic CAZALS** **President du jury**
Directeur de recherche, Inria, Sophia-Antipolis, France
**Dr. Dima KOZAKOV** **Rapporteur**
Research Associate Professor, Boston University, USA
**Prof. Dr. Raphaël GUEROIS** **Rapporteur**
Directeur de recherche, CEA, Saclay, France

I dedicate this Ph.D. dissertation to those with unquenchable source of belief and light for me - to my parents.

# Acknowledgements

I express sincere gratitude to my research guide Dr. Sergei Grudinin for the initiation of this Ph.D. followed by three years of teaching and patience, friendship and help. The joy and enthusiasm Sergei has for the research have been contagious and motivational for me, and I appreciate all the contributions of time, ideas, and funding to make my Ph.D. experience stimulating and productive. Work with Sergei has proceeded with interest and pleasure, minimizing tough times of my Ph.D. pursuit and making it free from any stress. Thank you so much for expanding my background in computational biology, for all these interesting projects we have investigated, and for my step-by-step development into an independent researcher. It has been an honor to be your first Ph.D. student.

I am especially grateful to my co-advisor Dr. Stephane Redon, the head of the Nano-D team where I made the research presented here. I very much appreciate his enthusiasm, intensity, willingness to do everything on the highest level. Stephane has encouraged me to always be organized and keep a broader view on scientific problems. In my attempts to understand the French cuisine, Stephane introduced to me the vine fondue in the first day of my arrival in Grenoble; apparently it became clear that I would enjoy the Ph.D. period a lot. Thank you for your interesting suggestions and useful feedback whenever I have needed it.

Completion of this doctoral dissertation would not have been possible without my advisor professor Dr. Anatoli Iouditsky. Anatoli is someone you will instantly like and never forget once you meet him. I acknowledge Anatoli for the extremely helpful conversations at Laboratoire Jean Kuntzmann. The ease and joy he narrates about complex mathematical objects inspired me to learn and implement different convex optimization methods used in this research.

For this dissertation I thank my reading committee members Dr. Dima Kozakov and professor Dr. Raphael Guerois as well as the other two members of my oral defense committee professor Dr. Frederic Cazals and Dr. Juan Cortes for their time, interest, helpful comments, and insightful questions. Thank you for letting my defense be an enjoyable moment. It has been my pleasure to see such bright scientists in the jury.

The members of Nano-D have contributed immensely to my personal and professional time at Inria. The team has been a source of friendships as well as good advice and col-

laboration comprising general and scientific talks, group seminars, coffee breaks, sport and board games, and I do hereby acknowledge all of the members. Dr. Svetlana Artemova and Dr. Mael Bosson have been my primary examples of Ph.D. students, with them I did my first ascending to the French mountains. Jocelyn Gate, a cheerful software engineer, has been ready to a joke at any time regardless the workload and the mood. Dr. Leonard Jaillet has been my first long-term office-mate and I am grateful for the chess talks, language exchange and his friendship. I acknowledge Krishna-Kant Singh for his open to discussion mind. Krishna is always eager to share and participate with his crusading zeal. I will miss our silliness and childish sessions and I still get a laugh when I reminisce about the time you opened 30-days-aging apple juice. I thank my other office-mates Zofia Trstanova and Khoa Nguyen. Zofia has been the limiting factor for us to maintain a shred of discipline in the office, and I am grateful to Khoa for always closing the door. I thank Mohammed Yengui for his peculiar character and price/quality selectivity of the restaurants. I acknowledge Emilie Neveu for friendly support and pleasant collaboration. I also thank Marc Piuzzi, Alexandre Hoffmann and new members of Nano-D with which I have not got an opportunity to really know them due to the tense ending of the dissertation, but they all have been very kind and personable to me: Simeho Edorh, Mark Aubert, Francois Rousse, and Nadhir Abdellatif. I acknowledge Elie Namias for his support and other interns rotating in Nano-D, who was friendly to me: Antoine Plet, Mathias Louboutin, Jelmer Wolterink, Georgy Cheremovskiy, Astha Agarwal, Himani Singhal, Maria Werewka, Gabriel Gonzalez, and Guillaume Pages.

I express my gratitude to our neighbours in the Minatec building, the Corse team at Inria, however, this name might be changed while I was writing this sentence. Dr. Fabrice Rastello, the team leader, is a person with an amicable and positive disposition. I am very much indebted to the friendly atmosphere created by Fabian Gruber, who is always in; Diogo Sampaio, whose apartment with the best cachacas is always open for friends; Alexandros Labrineas, whose friendship and conversations not related to science have been humorous, inspiring and useful.

I have had pleasure to communicate with or alongside of past and present Inria's members: Laurentiu Trifan, Darren Wraith, Francois-Xavier Boillot, Caroline Richard, Farida Enikeeva, Michel Amat, Bruno Roberto, Diana Stephane, Lukasz Domagala and many others.

I thank assistants Zilora Zouaoui, Helene Baum, Francoise De Coninck, Imma Presseguer, and all human resources for their administrative help. Especially I acknowledge Zilora for her irreplaceable help with the training modules required for the defense. And many-many-many thanks to Imma Preseguer for the help, support, enthusiasm, and energy. Imma is a source of smile and optimisme in Nano-D, and she can solve any problem that scientist

cannot.

I acknowledge Dr. Dave Ritchie and Dr. Dima Kozakov for the bright collaboration and shared experience. The DockTrina algorithm presented in this dissertation would not have been possible without Dave. Dima has helped with the comparison part of the CARBON algorithm also presented here. I appreciate your scientific advices and knowledge and many insightful discussions and suggestions. I thank Zaid Harchaoui and Roland Hildebrand for the helpful conversations related to the optimization part of this dissertation. I am also very grateful to the CAPRI assessment committee Shoshana Wodak, Sameer Velankar, Marc Lensink and all organizers of this primary resource for testing methods aimed to predict protein-protein structures.

I gratefully acknowledge the funding sources that made this Ph.D. work possible: National Funding Agency for Research and European Research Council.

This thesis represents not only my work, it is a milestone in more than one decade of work of my respected teachers and mentors. I appreciate work of my high school teachers in Irkutsk and Moscow. Thank you for the primary knowledge of mathematics, physics, informatics and other classes you gave to me. I especially thank Andrey Alexandrov and Marianna Gelfand, my teachers in math and physics. They were and remain my best role model for teachers.

Андрей Георгиевич и Марианна Давыдовна! Я вспоминаю вас и о времени огонька в глазах с улыбкой, теплотой и благодарностью. Спасибо за непереоценимый вклад в мое образование, ваше чувство юмора и настоящее искусство преподавать!

I also thank my teachers at Moscow Institute of Physics and Technology for their scholarly input I received throughout my study and teachers at M.M Shemyakin and Y.A. Ovchinnikov Institute of Bioorganic Chemistry for the fundamental background in life science they gave.

My stay in France have been made so enjoyable and colourful in large part grace to my friends and familiars, and I warmly thank all of you. Namely I acknowledge my university friends Dr. Vitaly Polovinkin, Dr. Ivan Gushchin, Anton Abyzov and especially my new friends Margarita Merkulova, Ekaterina and Igor Markelov. Together we took part in plenty of activities, hiked to a lot of mountains, visited many restaurants and I am grateful for time spent with you and for your support in the final stage of this Ph.D. I also thank friends in Russia, who have been always ready to meet me: Marina and Victor Yakovlev, Svetlana and Alexandra Ahmineeva, Yulya and Andrey Avramenko, Ludmila Starzhinskaya, Alexandr Tyulenev, Svetlana and Alexey Mishin, and Dmitriy Luchinkin. I especially acknowledge Ksenia Chekashkina whose sleepless nights, care, and help on the eve of the Ph.D. defense

is so appreciated.

If I have forgotten anyone, I apologize.

I am grateful to my parents for all their love and encouragement. They have cherished with me every great moment and unconditionally supported me whenever I needed it. I deeply miss my stepfather, who raised me with a love of science and who is very proud of me. I especially thank my mom:

Мама, поздравляю, что наконец-то у тебя появился ребенок с ученой степенью! Спасибо тебе, что у меня есть пример силы воли и любви, ты - моя стрелка "Вперед" в любые моменты жизни.

# Abstract

Drug discovery is a multidisciplinary field which includes molecular biology, biophysics, biochemistry, and pharmacology. It usually starts with the identification of a biological target which is known to play a critical role in a particular disease. Therein, computational methods are increasingly used in the structure-based drug design from target identification and validation to the designing of new molecules. To identify molecules that inhibit desirable activity hundreds of thousands candidates generated with docking protocols are virtually screened to filter out top-scoring hits. The latter are then tested in biological environment and many cycles of optimization are performed to obtain the candidates for further clinical trials. The first algorithm dedicated to the docking of small molecules was applied to find new candidates against HIV-1 protease in 1990. Since then, using of docking pipelines has become a standard practice in drug discovery.

Typically, a docking protocol comprises different phases. It starts with the exhaustive sampling of the binding site upon rigid-body approximation of monomers. Then, clustering algorithms are used in order to group similar binding candidates. Different refinement methods are applied in order to take into account flexibility of a molecular complex or to get rid of possible docking artefacts. Finally, binding candidates are scored with energy functions and top-ranked predictions are selected. The Thesis presents novel algorithms for docking protocols to facilitate structure prediction of protein complexes, which belong to one of the most important target class in the structure-based drug design.

First, *DockTrina* - a new algorithm to predict conformations of triangular protein trimers (i.e. trimers with pair-wise contacts between all three pairs of proteins) is presented. The method takes as input pair-wise contact predictions from a rigid-body docking program. It then scans and scores all possible combinations of pairs of monomers using a very fast root mean square deviation test. Finally, it ranks the predictions using a scoring function which combines triples of pair-wise contact terms and a geometric clash penalty term. Being fast and efficient, DockTrina outperforms state-of-the-art computational methods dedicated to predict structure of protein oligomers on collected benchmark of protein trimers.

Second, *RigidRMSD* - a C++ library which provides fast way to compute root mean

square deviations (RMSDs) between rigid-body transformations of a molecule is developed. The library is practically useful for clustering of docking poses, resulting in ten times speed up compared to standard RMSD-based clustering algorithms. The theoretical foundation of the RigidRMSD algorithm is also used in scoring and refinement stages of the docking pipeline.

Third, *KSENIA* - a novel knowledge-based scoring function for protein-protein interactions is developed. The problem of scoring function reconstruction is formulated and solved as a convex optimization problem in high-dimensional Euclidean space. As a result KSENIA is a smooth function and, thus, is suitable for the structure refinement with potential energy functions. Remarkably, it is shown that using information about only native interfaces of protein complexes is sufficient to reconstruct well-discriminative scoring function.

Fourth, *CARBON* - a new algorithm for the rigid-body refinement of docking candidates is proposed. The rigid-body optimization problem is viewed as the calculation of quasi-static trajectories of rigid bodies influenced by energy function. To circumvent the typical problem of incorrect step-sizes for rotation and translation movements of molecular complexes, the concept of controlled advancement is introduced. CARBON works well in combination with classical force-field and knowledge-based scoring function. It is a suitable tool for the rigid-body refinement of molecular complexes with moderate and large steric clashes between its monomers.

Finally, a novel method to evaluate prediction capability of scoring functions is introduced. It allows to rigorously assess the performance of the scoring function of interest on benchmarks of molecular complexes. The method manipulates with the score distributions rather than with scores of particular conformations, which makes it to be advantageous compared to the standard hit-rate criteria.

The methods described in the Thesis are tested and validated on various protein-protein benchmarks. The implemented algorithms are successfully used in the CAPRI contest for structure prediction of protein-protein complexes. The developed methodology can be easily adapted to the recognition of other types of molecular interactions, involving ligands, polysaccharides, RNAs, etc. The C++ versions of the presented algorithms will be made available as a SAMSON Element for the SAMSON software platform at http://www.samson-connect.net or at http://nano-d.inrialpes.fr/software/.

# Résumé

Le docking moléculaire est une méthode permettant de prédire l'orientation d'une molécule donnée relativement à une autre lorsque celles-ci forment un complexe. Le premier algorithme de docking moléculaire a vu jour en 1990 afin de trouver de nouveaux candidats face à la protéase du VIH-1. Depuis, l'utilisation de protocoles de docking est devenue une pratique standard dans le domaine de la conception de nouveaux médicaments. Typiquement, un protocole de docking comporte plusieurs phases. Il requiert l'échantillonnage exhaustif du site d'interaction où les éléments impliqués sont considérées rigides. Des algorithmes de clustering sont utilisés afin de regrouper les candidats à l'appariement similaires. Des méthodes d'affinage sont appliquées pour prendre en compte la flexibilité au sein complexe moléculaire et afin d'éliminer de possibles artefacts de docking. Enfin, des algorithmes d'évaluation sont utilisés pour sélectionner les meilleurs candidats pour le docking. Cette thèse présente de nouveaux algorithmes de protocoles de docking qui facilitent la prédiction des structures de complexes protéinaires, une des cibles les plus importantes parmi les cibles visées par les méthodes de conception de médicaments.

Une première contribution concerne le nouvel algorithme Docktrina qui permet de prédire les conformations de trimères protéinaires triangulaires (i.e. des trimères pour lesquels des interactions mutuelles de contact existent pour chacune des trois paires de protéines). Celui-ci prend en entrée des prédictions de contacts paire-à-paire à partir d'hypothèse de corps rigides. Ensuite toutes les combinaisons possibles de paires de monomères sont évalués à l'aide d'un test de distance RMSD efficace. Cette méthode à la fois rapide et efficace améliore l'état de l'art sur les protéines trimères.

Deuxièmement, nous présentons RigidRMSD une librairie C++ qui évalue en temps constant les distances RMSD entre conformations moléculaires correspondant à des transformations rigides. Cette librairie est en pratique utile lors du clustering de positions de docking, conduisant à des temps de calcul améliorés d'un facteur dix, comparé aux temps de calcul des algorithmes standards.

Une troisième contribution concerne KSENIA, une fonction d'évaluation à base de connaissance pour l'étude des interactions protéine-protéine. Le problème de la reconstruction de fonction d'évaluation est alors formulé et résolu comme un problème d'optimisation convexe.

Quatrièmement, CARBON, un nouvel algorithme pour l'affinage des candidats au docking basés sur des modèles corps-rigides est proposé. Le problème d'optimisation de corps-rigides est vu comme le calcul de trajectoires quasi-statiques de corps rigides influencés par la fonction énergie. CARBON fonctionne aussi bien avec un champ de force classique qu'avec une fonction d'évaluation à base de connaissance. CARBON est aussi utile pour

l'affinage de complexes moléculaires qui comportent des clashes stériques modérés à importants.

Finalement, une nouvelle méthode permet d'estimer les capacités de prédiction des fonctions d'évaluation. Celle-ci permet d'évaluer de façon rigoureuse la performance de la fonction d'évaluation concernée sur des benchmarks de complexes moléculaires. La méthode manipule la distribution des scores attribués et non pas directement les scores de conformations particulières, ce qui la rend avantageuse au regard des critères standard basés sur le score le plus élevé.

Les méthodes décrites au sein de la thèse sont testées et validées sur différents benchmarks protéines-protéines. Les algorithmes implémentés ont été utilisés avec succès pour la compétition CAPRI concernant la prédiction de complexes protéine-protéine. La méthodologie développée peut facilement être adaptée pour de la reconnaissance d'autres types d'interactions moléculaires impliquant par exemple des ligands, de l'ARN... Les implémentations en C++ des différents algorithmes présentés seront mises à disposition comme SAMSON Elements de la plateforme logicielle SAMSON sur http://www.samson-connect.net or at http://nano-d.inrialpes.fr/software/.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Basic Concepts of Protein Structure

Proteins form one of the most important target class in the structure-based drug design. This section briefly introduces the basic concepts of protein structure hierarchy. Proteins performs a myriad of activities and orchestrate most of the essential functions of a cell, such as structural, transport and regulatory functions. The fundamental assumption in protein science states that protein structure leads to protein function. To describe the protein structure, it is hierarchicaly represented by the four-level hierarchy: primary, secondary, tertiary, and quaternary protein structure, which are schematically represented on Figure 1.1.

Proteins are linear polymers of amino acids and the primary structure is the sequence of amino acids composing the protein. Amino acids are molecules that contain an amino group ($NH_2$), a carboxyl group (COOH), a hydrogen (H) attached to a central carbon atom $C_\alpha$, and a side chain (R) attached to the $C_\alpha$. Amino acids are distinguished by the R group, which confers specific chemical properties on it. In proteins, amino acids are referred to as amino acid residues, which are connected to each other via the peptide bond. Thus, one often refers to the protein as to the polypeptide.

The secondary structure concept is dedicated to describe the local conformation of a polypeptide chain. Two types of such conformations are found to be dominant among all known structures of proteins: $\alpha$-helix and $\beta$-sheets. The former has the coil shape with a period of 3.6 residues per turn and the latter is the pleated sheet which is formed by $\beta$-strands, that is stretches of several residues laterally connected to each other. Both types of the secondary structure are stabilized by hydrogen bonding interactions. Whereas in the $\alpha$-helix the hydrogen bond is formed between the carbonyl oxygen of each residue and the amide proton of the residue four positions ahead, in the $\beta$-sheets the hydrogen bonds are located between adjacent polypeptide chains. There are also other regular structures besides

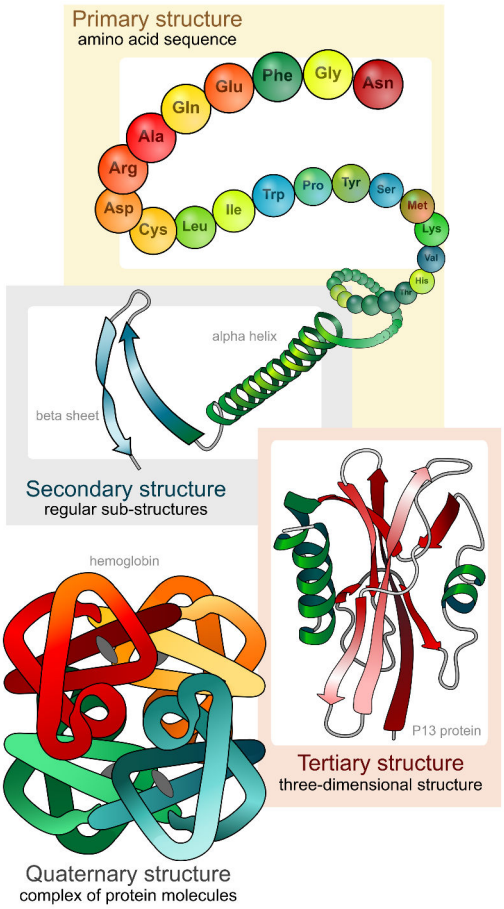


Figure 1.1 Schematic representation of four levels of protein structure hierarchy.

the standard $\alpha$-helix and $\beta$-sheets as well as the irregular structures, such as loops.

The secondary structural elements combine into three-dimensional structure of protein, called the tertiary (ternary) structure or fold of the protein. The process of tertiary structure formation is referred as to the protein folding. Within the overall protein fold one can distinguish structurally independent patterns (domains) and motifs - regions which could be structurally dependent. Domains and motifs often have functional significance, hence, they could be considered as the functional units. The tertiary structure is stabilized by the intramolecular interactions (between the structural elements) and the intermolecular forces (with surrounding molecules, e.g. solvent). The folds are classified based on the biochemical principles (globular, membrane and fibrous) as well as on the evolutionary principles and structural organization [4, 129].

Whereas the tertiary structure describes a single polypeptide chain (monomer), the quaternary structure is dedicated to oligomeric protein complexes (multimer), which consist of two or more interacting monomers. When the monomers have the same tertiary structure

the protein complex is called homomeric, and it is called heteromeric otherwise. Albeit the quaternary structure is stabilized by very specific interactions, they are of the same type as those employed in tertiary and secondary structure formation. The association of monomers into the multimers provides several functional advantages. For example, monomers bound together enhance binding capabilities of the multimer compared to the individual monomer; upon their association, monomers can confer multiple functions on a single protein; protein function can be altered when the quaternary structure is changed due to the combinatorial shifts, that is swapping of different monomers; finally, formation of large protein complexes is possible thanks to the association of large number of small monomers.

Throughout the Thesis particular protein complexes are referred as to its identification numbers in the Protein Data Bank (PDB) [13]. The PDB database contains structure of proteins determined using the X-ray crystallography, nuclear magnetic resonance, and cryo-electron microscopy. Albeit modern experimental techniques are very efficient in protein structure prediction, it is still difficult to determine quaternary structures of protein complexes due to its size, flexibility and other factors. Thus, computational methods dedicated for the protein structure prediction serve as a faster and cheaper alternative to the experimental techniques.

## 1.2 Protein Structure Prediction with Docking Pipelines

The docking pipelines are designated to predict the structure of a molecular complex from the structures of its individual subunits [144]. In case of protein docking, one distinguishes between the protein-ligand and the protein-protein docking. The former often implies that the ligand is a small chemical compound and the binding site is often predetermined. The latter typically deals with two protein molecules and the global search of the binding pose is required. Similarly, the smaller protein is often called the ligand and the the bigger protein is called the receptor. The main part of a docking protocol comprises the exhaustive searching of the binding site upon rigid-body approximation of proteins, that is the global search of favourable orientations of the receptor with respect to the ligand in six rigid-body translational and rotational degrees of freedom. Thanks to the pioneering work of Katchalski-Katzir et al. [62] most of modern docking algorithms employ the fast Fourier transform (FFT) to perform the global search of the binding poses [143]. Nonetheless, the structure of monomers outside the complex (unbound state) often undergo conformational changes upon binding, resulting in the bound state conformation. The modeling of the unbound to bound transition is a very challenging task, because this transition typically involves much more degrees of freedom than six rotational and translational ones. To take into account protein

flexibility, many different refinement methods have been developed. Particularly, appropriate modeling of side chain conformation changes was shown to be efficient for solving most of docking cases [8, 65, 75, 144]. Molecular docking algorithms typically produce thousands of predictions, some of them having a very similar geometry. Therefore, it is practical to group these into clusters and consider only one representative binding candidate from each cluster. There are multiple ways to measure similarity between molecular structures [147], however, most of the modern state-of-the-art clustering algorithms use the pair-wise root mean square deviation (RMSD) as the similarity metrics between the predictions.

The docking candidates are ranked with respect to the energy (or the score) given by the scoring function implemented in the docking algorithm. Currently, FFT docking involves terms representing shape complementarity, electrostatic, and desolvation contributions to assess the energy of the produced candidates. However, one is interested in the binding free energy:

$$\Delta G_{\text{bind}} = \Delta H - T\Delta S, \tag{1.1}$$

where $\Delta H$ is the enthalpic difference between the bound and the unbound states of the complex, $T$ is the temperature, and $\Delta S$ is the entropic difference upon binding. The binding free energy is a much more sophisticated function compared to the potential energy, and involves not only interaction energy between the partners, but also changes in the internal energy of monomers, interactions with solvent, rearrangement of solvent molecules and changes of conformational degrees of freedom corresponding to the entropic loss upon binding. Direct computation of the binding free energy of proteins is an intractable problem due to its high computational cost. Instead, different scoring functions as an approximation to the binding free energy have been extensively developed to be applied to putative docking poses and virtual screening candidates. The scoring functions for the virtual screening and the selection of putative binding poses can be categorized into three groups: first-principle methods, empirical scoring functions, and knowledge-based potentials. First-principle methods generally do not take into account solvent and intramolecular interactions, and compute only enthalpic interactions between the receptor and the ligand using molecular mechanics force-fields. Empirical scoring functions consist of a linear combination of terms that are known to reflect important factors of binding, e.g. hydrophobic contacts, hydrogen bonding, accessible and buried surface area, etc. Using regression models, all terms are supplied with the corresponding weights to provide a good agreement with the training set of complexes for which experimentally determined binding constants are available. Knowledge-based potentials are developed based on the structural information from databases of molecular complexes. The main assumption behind these potentials is that the native molecular complexes possess distinct structural features with respect to the non-native structures. For example,

Table 1.1 Quality of a docking prediction with respect to the $L_{\mathrm{RMSD}}$ value. The $L_{\mathrm{RMSD}}$ value is defined as the RMSD of the backbone atoms of the ligand after the receptors in the native and the docking pose conformation have been optimally superimposed.

| Quality | $L_{\mathrm{RMSD}}$ (Å) |
|---|---|
| 1 | $L_{\mathrm{RMSD}} \leq 1$ |
| 2 | $1 < L_{\mathrm{RMSD}} \leq 5$ |
| 3 | $5 < L_{\mathrm{RMSD}} \leq 10$ |

the common assumption is that more frequently observed interactions are more important for the complex stability. These potentials are generally more computationally efficient and less time demanding, and hence are suitable for the docking protocols.

At this point, energy and score are used as the synonyms. Since the original docking predictions usually do not possess optimal energy values, in practice, in order to achieve more reliable energy values, one performs refinement using the scoring function. Typically, rigid-body minimization algorithms are applied on the first step, where the energy of a conformation is minimized with respect to the rigid transformations of receptor and ligand [92].

Various docking, clustering, refining, and scoring algorithms are combined into various docking protocols. To assess prediction capabilities of the docking protocols, the Critical Assessment of PRedicted Interactions (CAPRI) was organized [55]. It is a blind prediction experiment, where the target is the experimentally established but unpublished structure of a protein-protein complex. Given structural information (primary, secondary or tertiary structures) about a receptor and a ligand in their unbound form, CAPRI tests docking algorithms to predict the bound structure of the complex. Additionally, CAPRI involves the scoring contest, when the participants are invited to rank all submitted models according to their scoring functions. Finally, the models are evaluated against the true target structure and the performance of docking and scoring protocols is reported.

To evaluate the relevance of a model, one introduces quality criteria; the two basic ones which are used through the Thesis are the ligand-RMSD and the CAPRI criterion. The ligand-RMSD ($L_{\mathrm{RMSD}}$), is defined as the value of RMSD of the backbone atoms of the ligand after the receptors in the native and the decoy conformations have been optimally superimposed (see Table 1.1). In the CAPRI contest, a more sophisticated criterion compared to the ligand-RMSD is used. More precisely, in addition to the ligand-RMSD, it involves the fraction of native contacts in the docking prediction $f_{\mathrm{nat}}$, and the interface RMSD, $I_{\mathrm{RMSD}}$ (see Table 1.2). The $f_{\mathrm{nat}}$ parameter is the ratio of the number of native residue-residue contacts in the predicted complex to the number of residue-residue contacts in the crystal

Table 1.2 The CAPRI criterion to estimate the quality of docking predictions

| Quality | Condition |
|---|---|
| 1 (high) | $f_{\text{nat}} \geq 0.5$ and ($L_{\text{RMSD}} \leq 1.0$ or $I_{\text{RMSD}} \leq 1.0$) |
| 2 (medium) | ($0.3 \leq f_{\text{nat}} < 0.5$) and ($L_{\text{RMSD}} \leq 5.0$ or $I_{\text{RMSD}} \leq 2.0$) or ($f_{\text{nat}} \geq 0.5$ and $L_{\text{RMSD}} > 1.0$ and $I_{\text{RMSD}} > 1.0$) |
| 3 (acceptable) | ($0.1 \leq f_{\text{nat}} < 0.3$ and ($L_{\text{RMSD}} \leq 10.0$ or $I_{\text{RMSD}} \leq 4.0$) or ($f_{\text{nat}} \geq 0.3$ and $L_{\text{RMSD}} > 5.0$ and $I_{\text{RMSD}} > 2.0$) |

structure. A pair of residues from different monomers are considered to be in contact if they are within 5 Å from each other. The $I_{\text{RMSD}}$ parameter is the RMSD of the interface region between the predicted and native structures after optimal superimposition of the backbone atoms of the interface residues. A residue is considered as the interface residue if any atom of this residue is within 10 Å from the other partner.

## 1.3 Contribution of the Thesis

### 1.3.1 DockTrina: Docking of Triangular Trimers

The first contribution concerns the problem of reconstruction of oligomeric protein complexes, particularly the docking of monomers forming three-multimer. In spite of the abundance of oligomeric proteins within a cell, the structural characterization of protein–protein interactions is still a challenging task. In particular, many of these interactions involve heteromeric complexes, which are relatively difficult to determine experimentally. Hence there is growing interest in using computational techniques to model such complexes. However, assembling large heteromeric complexes computationally is a highly combinatorial problem. Nonetheless the problem can be simplified greatly by considering interactions between protein trimers. After dimers and monomers, triangular trimers (i.e. trimers with pair-wise contacts between all three pairs of proteins) are the most frequently observed quaternary structural motifs according to the three-dimensional (3D) complex database. The first contribution of the Thesis comprises ***DockTrina [108]*** - a novel protein docking method for modeling the 3D structures of nonsymmetrical triangular trimers. The method takes as input pair-wise contact predictions from a rigid-body docking program. It then scans and scores all possible combinations of pairs of monomers using a very fast root mean square deviation test. Finally, it ranks the predictions using a scoring function which combines triples of pair-wise contact terms and a geometric clash penalty term. The overall approach takes less than 2 min per complex on a modern desktop computer. The method is tested and

validated using a benchmark set of 220 bound and seven unbound protein trimer structures.

### 1.3.2 Rapid Determination of RMSDs corresponding to Macromolecular Rigid-body Motions

The second contribution concerns finding the RMSDs between two coordinate vectors that correspond to the rigid-body motion of a macromolecule, which is an important problem in structural bioinformatics, computational chemistry, and molecular modeling. Standard algorithms compute the RMSD with time proportional to the number of atoms in the molecule. However, using the rigid-body formalism, the RMSD could be computed more efficiently, resulting in a fast and efficient approach. Thus, the second contribution of the Thesis comprises ***RigidRMSD [104]***, a new algorithm that determines a set of RMSDs corresponding to a set of rigid-body motions of a macromolecule in constant time with respect to the number of atoms in the molecule. The algorithm is particularly useful for rigid-body modeling applications, such as rigid-body docking, and also for high-throughput analysis of rigid-body modeling and simulation results, e.g. clustering of the docking predictions. The theoretical foundation of the RigidRMSD algorithm is also used in scoring and refinement stages of the docking pipeline.

### 1.3.3 Knowledge-based Scoring Function for Protein-Protein Interactions

The third contribution concerns selection of putative binding poses, which is a challenging part of virtual screening for protein-protein interactions. Predictive models to filter out binding candidates with the highest binding affinities comprise scoring functions that assign a score to each binding pose. Existing scoring functions are typically deduced collecting statistical information about interfaces of native conformations of protein complexes along with interfaces of a large generated set of non-native conformations. However, the obtained scoring functions become biased toward the method used to generate the non-native conformations, i.e. they may not recognize near-native interfaces generated with a different method. It is demonstrated that knowledge of only native protein-protein interfaces is sufficient to construct well-discriminative predictive models for the selection of binding candidates. More precisely, a new scoring method is introduced – it comprises a knowledge-based potential called ***KSENIA [105]*** deduced from the structural information about the native interfaces of 844 crystallographic protein-protein complexes. KSENIA is derived using convex optimization with a training set composed of native protein complexes and their

near-native conformations that are obtained using deformations along the low-frequency normal modes. As a result, the knowledge-based potential has no bias toward a method to generate putative binding poses. Furthermore, KSENIA is smooth by construction, which allows to use it along with a rigid-body optimization to refine the binding poses. Using several test benchmarks it is demonstrated that the new method discriminates well native and near-native conformations of protein complexes from the non-native ones.

### 1.3.4 CARBON: Controlled-Advancement Rigid-Body Optimization for Nanosystems

The fourth contribution of the Thesis comprises a fast and efficient method for the rigid-body refinement of molecular complexes, called ***CARBON [107]***, where we consider the rigid-body optimization problem as the calculation of quasi-static trajectories of rigid bodies influenced by the inverse-inertia-weighted energy gradient. In order to determine the appropriate step size in the direction of the net generalized force, we introduce the concept of advancement region, which is the interval of step sizes that provide movements of the rigid body within a certain range of RMSD from the initial conformation. As a result, the CARBON approach guarantees the absence of incorrectly large movements of the rigid-bodies as well as the absence of irrelevantly small movements. CARBON is tested and validated on several benchmarks using both a classical force-field and a knowledge-based scoring function. It is demonstrated that CARBON significantly improves the quality of docking predictions, resulting in higher success rate of the scoring protocol. Finally, CARBON remains stable when monomers of a molecular complex significantly overlap and efficiently resolves moderate and large steric clashes.

### 1.3.5 A Novel Criterion to Evaluate Scoring Power of Scoring Functions for Molecular Complexes

Efficiency of scoring functions is typically assessed using benchmarks that comprise many non-native conformations (decoys) and a few near-native conformations, both obtained with docking algorithms [94]. As a result, a single scoring function could demonstrate a different scoring power on benchmarks based on the same set of native complexes but with decoys generated with different docking algorithms. Furthermore, the fact that a scoring function can/cannot able to distinguish *one* particular near-native candidate does not imply that it can/cannot distinguish *any* near-native candidate. Thus, the scoring power is a strongly biased criterion, which critically depends on the poses of the binding candidates in the bench-

mark set. To address the latter problem, we introduce an alternative criterion to evaluate the scoring power of a scoring function, which is free of the above-mentioned disadvantages ***[106]***. More precisely, we complement the benchmark set with the constructed uniform ensembles of near-native conformations, where each conformation lies within a certain RMSD from the corresponding native conformation. We provide the fast and efficient method to generate the uniform ensembles of near-native conformations. Then, we estimate the scoring power of a scoring function using the cumulative distribution function of decoy scores and the probability density function of the near-native conformation scores.

The method was applied to assess the scoring power of the knowledge-based scoring functions for the protein-protein complexes, which we derive using the modern convex optimization apparatus. Particularly, the obtained results indicate that the derived scoring function discriminate well conformations within 2 Å, but performs poorly for the conformations of 5 Å

The methods described in the Thesis are tested and validated on various protein-protein benchmarks. The implemented algorithms are successfully used in the CAPRI contest for structure prediction of protein-protein complexes. The developed methodology can be easily adapted to the recognition of other types of molecular interactions, involving ligands, polysaccharides, RNAs, etc. The C++ versions of the presented algorithms will be made available as SAMSON Elements for the SAMSON software platform at http://www.samson-connect.net or at http://nano-d.inrialpes.fr/software/.

# Chapter 2

# DockTrina: Docking of Triangular Trimers

## 2.1 Introduction

Most proteins interact with other proteins. They form protein complexes that are essential for many biological processes and which are responsible for a vast array of biological functions [43]. It has been shown that the human genome encodes around 30,000 proteins which are involved in about 130,000 protein-protein interactions [15]. A recent study of 2000 yeast proteins found that more than 80% of the proteins interact with at least one other protein [38]. Furthermore, about 50% of them form complexes with more than five other partners [1]. These data clearly indicate the importance of protein interactions within oligomers in a cell.

In spite of the abundance of oligomeric proteins, the structural characterization of protein–protein interactions is still a challenging task. For example, monomeric structures constitute more than 50% of the structures in the PDB database [13], whereas only about 30% of the protein sequences in SwissProt [83] correspond to monomeric structures [78]. This disparity reflects the relative difficulty of determining the structures of oligomeric proteins experimentally.

Undoubtedly, homomeric interactions are the most common type of protein–protein interactions [40, 78, 101]. Indeed, between about 50% and 70% of the protein complexes of known three-dimensional (3D) structure are homooligomers according to the SwissProt [83]. and Protein Quaternary Structure (PQS) databases [44]. Among the complexes that are not pure homo-oligomers, about two thirds contain at least one interaction between identical chains, and about one third involve purely heteromeric interactions [101].

Levy et al. [78] classified protein complexes of known 3D structure using a hierarchical graph representation of their fundamental structural features. They demonstrated that triangular trimers (i.e., trimers with pair-wise contacts between all three pairs of proteins) are the most frequent quaternary structure motif in the PDB after dimers and monomers. Particularly, triangular trimers constitute about two thirds of all trimers in the PDB, whereas only one third of trimers are linear trimers. Levy et al. [78] also demonstrated that triangular trimeric motifs appear in many other oligomers. For instance, if one considers oligomers of order less than 10 (accounting for more than 98% of all PDB structures), 67 out of 97 of these topologies contain triangular trimer motifs. Also, although two thirds (2044/3236) of Levy's set of 3236 nonredundant protein complexes are monomers and dimers, more than half of the remainder (i.e., > 536/1192) involve triangular protein interactions. On the other hand, several algorithms have been developed to predict symmetrical homomeric protein complexes, whereas only a very few are able to predict nonsymmetrical oligomers.

### 2.1.1 Docking of Symmetrical Protein Complexes

Here, we review seven published methods to perform docking of symmetrical protein complexes. Most of these methods can handle different types of symmetry, and some can model conformational changes within monomers. One of the methods for $C_n$ cyclic symmetry docking is SymmDock [123]. It demonstrated good performance for 19 bound complexes for which the monomers are related by noncrystallographic symmetry (NCS). The SymmDock algorithm consists of four steps. First, it computes a sparse dot surface representation for each protein, giving about six surface points for each surface atom. It then samples possible symmetry axes for each pair of surface points using geometric constraints provided by $C_n$ symmetry. Next, it clusters candidate symmetry axes according to their directions and the projection of the protein's center of mass onto the axes. Finally, it ranks the clusters by the number of matched pairs of surface atoms. If additional experimental data are available, for example, from small angle X-Ray scattering (SAXS) experiments, one may improve SymmDock results using FoXSDock [122]. This algorithm filters predicted models according to SAXS profiles, and it clusters and refines the interfaces using flexible docking with FireDock [87].

Mashiach-Farkash et al. [88] recently developed SymmRef to refine candidate docking solutions from any symmetric docking method. SymmRef models both side-chains and backbone movements and re-ranks the models by an energy scoring function, which includes various energetic terms. SymmRef does not apply symmetry constraints at the side-chain level. It was tested on an unbound set of 16 proteins.

The ROSETTA program [118] uses a Monte Carlo-plus-minimization protocol which

can deal with cyclic, dihedral, helical, and icosahedral symmetries [2]. It uses real-space minimization to find the lowest energy conformation of binding partners restricted by a certain type of symmetry constraint. The protocol consists of a low-resolution search using a residue-level potential followed by a subsequent high-resolution stage with all-atom energy function [41]. Symmetry is also used to restrict the number of backbone and side-chain degrees of freedom. The method was tested on 14 bound-bound cyclic oligomers related by NCS, and on one helical and one icosahedral complex.

Huang et al. [49] exploit the redundancy associated with $C_2$ cyclic symmetry for fast Fourier transform (FFT)-based docking for homodimers. More precisely, they reduce the rotational search space from $2\pi \times \pi \times 2\pi$ to $\pi \times \pi$. They tested their method on 121 bound complexes of homodimers collected by Bahadur et al. [5]. In a similar manner, M-ZDOCK can rigidly dock monomers to make $C_n$ symmetrical multimers [102]. To do this, it exhaustively explores the search space with two rotational and two translational degrees of freedom. The translational search is accelerated using a 2D FFT. Its scoring function includes surface complementarity as well as desolvation and electrostatic energy terms. M-ZDOCK was tested on a benchmark of eight bound/quasi-bound protein complexes and four unbound protein complexes with $C_n$ symmetry.

The ClusPro program can deal with $C_n$, $D_2$, and $D_3$ symmetries [27]. It uses the DOT rigid-body docking program [85] to generate and cluster more than 20,000 docked conformations between two monomers. It then selects the best 2,000 complexes according to its scoring function. Then, it constructs symmetrical $N$-oligomers and computes the root mean square deviation (RMSD) between the first monomer and its $N+1$ symmetrical replica. Finally, it filters out predictions with the RMSD beyond a certain threshold. The method was tested on 107 homo-oligomer complexes, including dimers, trimers, tetramers, pentamers, and hexamers.

Another geometric docking algorithm for $C_n$ and $D_n$ symmetries was described by Berchanski et al. [10, 11] . It is based on filtering binary homodimer docking solutions with symmetrical restrictions. First, the algorithm constructs homodimers using the MolFit docking program [62]. It then identifies symmetry-related homodimers by applying Euler's theorem to the docking transformation matrices (Euler's theorem states that the rotation angle $\phi$ about the eigenvector of a rotational matrix $M$ can be calculated from its trace: $Tr(M) = 2\cos(\phi) + 1$). Finally, this algorithm assembles constructed homodimers into $C_n$ or $D_n$ oligomers using geometric considerations. It was tested on eight $C_n$ and three $D_n$ symmetrical complexes.

### 2.1.2 Docking of Nonsymmetrical Protein Complexes

In this section, we review five methods able to predict nonsymmetrical protein oligomers. The CombDock program uses a combinatorial assembly approach to predict the 3D structures of nonsymmetrical protein oligomers and multi-protein complexes [54]. The algorithm starts from multiple pair-wise docking poses generated by the authors' original geometric hashing algorithm. Given as input $N$ protein structures, it first computes $N(N-1)/2$ sets of pairs of contacts between the proteins. Then, using at most 100 best contacts for each pair of proteins, CombDock builds an edge-weighted multigraph in which each protein is represented by a vertex, and each transformation that potentially docks a pair of proteins is represented by an edge between the corresponding vertices. The search for the best combination of edges uses the notion of spanning trees (a spanning tree is an undirected acyclic graph in which each vertex is visited exactly once). Because the problem of finding the minimum weight spanning tree is known to be NP-Hard, CombDock uses a greedy breadth-first strategy to assemble feasible (clash-free) subtrees into candidate solutions. It then clusters the generated solutions by RMSD, and it ranks them using a scoring function which includes both geometric and chemical terms. The method was validated on four bound and five unbound complexes.

Kim and Hummer [64] developed a method that performs oligomeric docking using coarse-grained models of individual monomers. The method uses Monte Carlo simulations with subsequent distance-based clustering between pairs of docking solutions. The oligomeric solutions are ranked according to their cluster populations. The feasibility of this method was demonstrated on the Vps27-ubiquitin complex in the presence of a membrane.

The HADDOCK multibody docking algorithm takes a more general approach for molecular docking [60], which can incorporate experimental and bioinformatics data to drive the modeling process. In HADDOCK, docking can be driven by a variety of experimental information about the interface, contacts, and relative orientations inside a complex. Furthermore, HADDOCK treats this information simultaneously and can deal with arbitrary symmetry through the use of user-defined distance constraints. The method performs energy minimization in dihedral angle subspace. Its energy function combines various energetic terms with user-defined ambiguous interaction restraints to favor the appearance of experimentally observed interactions and optional symmetry constraints. Although the authors tested HADDOCK only on five symmetrical homo-oligomeric proteins and one symmetrical protein-DNA complex, the method can be used to predict nonsymmetrical complexes with up to six chains.

ATTRACT [150] is a coarse-grained pair-wise docking algorithm with reduced amino

acid representation. This coarse-grained approach allows an efficient multistart search by energy minimization to be used in several directions to simulate global flexibility. Each direction corresponds to a soft collective degree of freedom computed with the normal mode analysis. Local flexibility is also included in ATTRACT by means of a mean-field representation of small loops and side chains. Although ATTRACT was not designed to predict trimers, the PTools library [120] may be used to call ATTRACT to predict trimeric complexes. The authors of PTools demonstrated its ability to find the correct structure of a symmetrical trimer complex without using symmetry information.

Recently, the Multi-LZerD algorithm for predicting multimeric complexes was developed [32]. In the first step, Multi-LZerD generates pair-wise docking predictions by applying the geometric hashing technique, where the protein surface is represented using Zernike–Canterakis basis functions. Then, it builds the spanning tree representation, with a node corresponding to a protein chain and an edge between nodes corresponding to a decoy of two chains. Similar to CombDock, which also uses spanning trees, Multi-LZerD implements a stochastic search genetic algorithm to find the solutions which are then ranked using a physics-based score. Finally, after clustering the predictions, Monte Carlo energy minimization strategy refines each complex in the final population. Multi-LZerD was tested on eleven bound and a few unbound multimeric complexes.

### 2.1.3 The DockTrina Approach

The surprisingly large number of non-redundant trimeric complexes found by Levy et al. [78] motivated us to develop the DockTrina algorithm for predicting new trimeric structures. As will be shown below, the problem of modeling triangular trimers can be solved very efficiently. DockTrina takes as input a set of pair-wise contact predictions for each pair of proteins in the trimer. In principle, any pair-wise rigid body docking algorithm could be used, but here we use the Hex program [115] because it is fast and because it can output the spatial transformations required by DockTrina. Given a set of pair-wise docking interactions, DockTrina exhaustively scans all possible combinations of contacts between three monomers (which typically involves evaluating some $10^{10}$ combinations), and it filters out any combinations which do not satisfy a very efficient RMSD test. This test is performed in constant time, which makes our method extremely fast. Finally, DockTrina ranks the predictions using a scoring function that combines the three pairwise docking scores with an empirical geometric penalty term. The typical running time of DockTrina is less than 2 minutes on a desktop computer with a 12-core 2.67-GHz Intel Xeon CPU.

We demonstrate the efficiency and accuracy of the method using a benchmark set of 220 protein trimers taken from bound crystal structures. This benchmark includes 85 proteins

with crystallographic symmetry, 76 protein complexes related by NCS, and 59 nonsymmetrical protein complexes. We also validate DockTrina using a further set of seven protein trimers involving unbound crystal structures. In addition, we compare the performance of DockTrina with SymmDock, CombDock, and HADDOCK algorithms.

## 2.2 Methods

### 2.2.1 The DockTrina Algorithm

Here, we describe the general work-flow of our algorithm. Given three proteins, A, B, and C, of a trimer, we first identify candidate contacts for the native interfaces between each pair of proteins: A–B, B–C, and C–A. For each pair-wise contact, DockTrina requires a rigid-body transformation $T$ obtained with Hex or other rigid-body docking algorithms, and a pseudo-energy score such that if the initial coordinates of the two proteins are $A^0$ and $B^0$, each putative docking solution is given by $A^0$ and $B = T^{AB} \cdot B^0$, as shown in Figure 2.1 A. Given a set of such putative pair-wise contacts, we then exhaustively evaluate triples of all possible combinations of such pairs.

More precisely, given a set of pair-wise docking transforms $T^{AB}$, $T^{BC}$, and $T^{CA}$, we form a tetramer of proteins A, B, and C, positioned at $A$, $B$, $C$, and $A'$, where $A'$ is the position of protein A after the above three transforms have been applied to it. This is shown in Figure 2.1 B. In other words, we initialize DockTrina with three proteins positioned at $A^0$, $B^0$, and $C^0$, such that the first protein is bound to the reference frame of the tetramer, and thus $A = A^0$. We can then observe that in this reference frame, the position of protein B is given as $B = T^{AB} \cdot B^0$. Similarly, the position of protein C is given as $C = T^{AB} \cdot T^{BC} \cdot C^0$, and the transformed position of the first protein A is given as $A' = T^{AB} \cdot T^{BC} \cdot T^{CA} \cdot A$. Therefore, we compute all possible combinations of three transforms of the following form:

$$T^{AA} = T^{AB} \cdot T^{BC} \cdot T^{CA}, \tag{2.1}$$

such that $A' = T^{AA} \cdot A$.

If the individual transforms are perfectly mutually consistent, then protein A should be transformed into itself. Thus, for any given combination of pair-wise docking transformations, the cumulative transformation $T^{AA}$ intuitively corresponds to the mismatch in the position of protein $A$ after the triangular docking attempt (see Fig. 2.1 B). Hence, the quality of a trimer produced by this algorithm can be characterized by the RMSD between the initial position of protein A and its transformed position $A' = T^{AA} \cdot A$. We compute this RMSD in constant time as described in the Chapter 3. Then, we remove docking predictions with

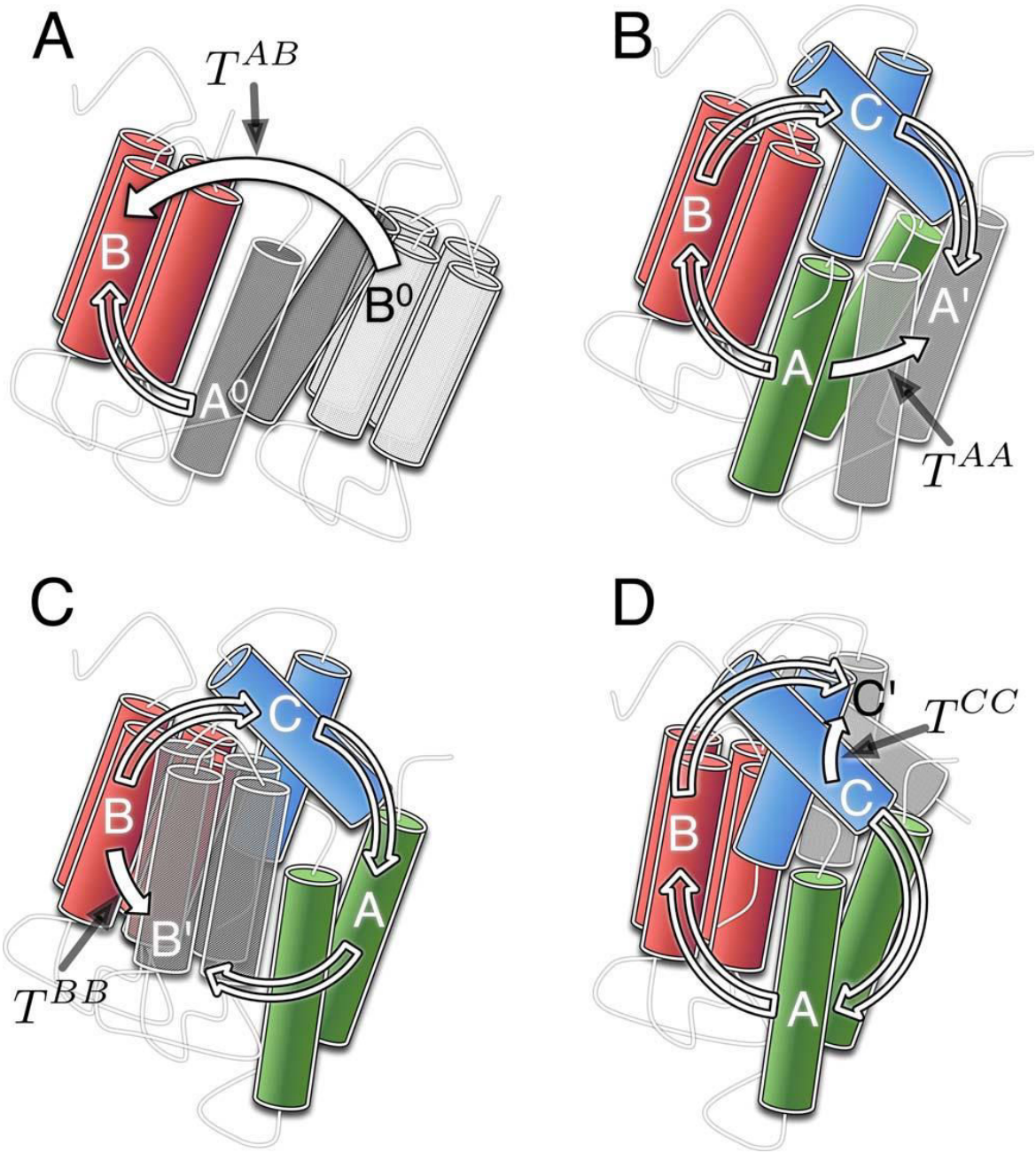


Figure 2.1 (**A** ) Illustration of a pair-wise docking algorithm, for example, Hex. Given the initial positions of a receptor $A^0$ and a ligand $B^0$, the algorithm generates a candidate transform $T^{AB}$ such that the predicted ligand position $B$ is given by $B = T^{AB} \cdot B^0$. (**B** , **C** , **D** ) Illustrations of the DockTrina trimer assembly algorithm. (**B** ) Given a set of transforms $T^A B$, $T^B C$, and $T^C A$, DockTrina forms a tetramer of proteins positioned at $A$, $B$, $C$, and $A^0$. Here, $A^0$ is the position of protein A after the application (in the given order) of the above three transforms. In the ideal case, $A^0$ should exactly superpose $A$. In practice, the quality of the A-B-C trimer is characterized by the mismatch between the $A$ and $A^0$ structure positions. (**C** , **D** ) The same procedure is repeated for tetramers of proteins positioned at $B$, $C$, $A$, $B^0$ and $C$, $A$, $B$, $C^0$ respectively.

the corresponding RMSD $> 10.0$ Å. We then assess the quality of the remaining predictions using the following scoring function:

$$\begin{aligned} \text{Score} &= \text{Score}^{AB} + \text{Score}^{BC} + \text{Score}^{CA} \\ &+ 0.25\frac{\text{Score}^{max}}{\text{RMSD}}, \end{aligned} \tag{2.2}$$

where $\text{Score}^{AB}$, $\text{Score}^{BC}$, and $\text{Score}^{CA}$ are the individual pair-wise scores, $\text{Score}^{\max}$ is the sum of three best scores for each contact, and RMSD (in Angstroms) is measured between the initial position $A$ and the transformed position $T^{AA} \cdot A$. We use an empirical penalty term $0.25\text{Score}^{\max}/\text{RMSD}$ to penalize large RMSDs. Finally, we remove predictions with the score $< 75\%$ of the sum of three best scores for each contact.

For a trimer of monomers A, B, and C, we need to apply the above procedure three times, with each monomer in turn providing the reference frame. Figure 2.1 (B–D) illustrates these steps. First, we do the calculations for the tetramer A–B–C–A$'$ computing the mismatch corresponding to the transform $T^{AA}$. Then, we repeat the calculations for the tetramers B–C–A–B$'$ with the transform $T^{BB}$ and C–A–B–C$'$ with the transform $T^{CC}$. These steps are summarised in Table 2.1. In the current version of our algorithm, we pro-

Table 2.1 The DockTrina algorithm.

1. Initialize RMSD computations with proteins A, B, and C.
2. Given a set of $M$ candidate contacts for each pair of proteins, compute:
   - $M^3$ spatial transformations $T^{AA}$, if proteins A, B, and C are identical;
   - $6M^3$ spatial transformations $T^{AA}$, $T^{BB}$, $T^{CC}$, $T^{-AA}$, $T^{-BB}$, $T^{-CC}$, if proteins A, B, and C are not identical and the transforms are slightly different when calculating pair-wise contacts for A-B compared to those for B-A;
   - $3M^3$ spatial transformations $T^{AA}$, $T^{BB}$, $T^{CC}$ otherwise.
3. Compute the RMSD for each of $M^3$, $3M^3$, or $6M^3$ spatial transformations.
4. Rescore predictions with RMSD values $\leq 10.0$ Å .
5. Sort predictions.

cess $M = 1{,}000$ pair-wise contacts. Therefore, for symmetrical structures, we evaluate $10^9$ different combinations of contacts for each target trimer. For NCS and nonsymmetrical target structures, we evaluate $6 \times 10^9$ different combinations of contacts. The multithreaded calculations for one NCS or nonsymmetrical protein take about 1.5 minutes on a desktop

machine running 64-bit Linux Fedora operating system with 12-core Intel(R) Xeon(R) CPU X5650 @ 2.67GHz. The corresponding multithreaded calculation for a symmetrical trimer takes about 15 seconds.

### 2.2.2 Pair-wise Docking with Hex

In principle, DockTrina may be used with any pair-wise docking algorithm that can output the spatial transformations described above. We chose to use Hex [115], to generate the necessary input transformations for DockTrina. Hex uses a polar Fourier representation of protein shapes, and it performs shape-based rigid-body docking using multiple FFTs to cover the 6D search space. Here, we used polar Fourier shape expansions to polynomial order $N = 31$. The real-space angular search step was 7.5°. We used the radial search range of 40 Å with a translational step of 2.5 Å and a subsequent substep of 1.2 Å. We clustered the docking solutions with a threshold of 8 Å and kept the best rigid-body spatial transforms for the first 1,000 of clusters. For each pair-wise docking solution, Hex outputs a calculated pseudo-energy and a spatial transformation to rotate and translate the "ligand" protein. However, because Hex uses discrete sampling rather than energy minimization, it can give slightly different predictions when calculating pair-wise contacts for A–B compared to those for B–A. Therefore, if the three monomers to be docked are nonidentical, we apply our calculations to the tetramers A–C–B–A′ , B–A–C–B′ , and C–B–A–C′. Thus, in such cases, we repeat the calculations six times. Thanks to multithreading, Hex needs only 3.3 minutes to obtain required transformations for the largest trimeric complex in our benchmark and 1.5 minutes for the smallest one.

### 2.2.3 The DockTrina Benchmark Sets

To test and validate the DockTrina docking algorithm, we constructed a benchmark set of protein trimers. This consists of 85 symmetrical trimers (including protein structures solved in H 3 and P 3 2 1 space groups), 76 NCS trimers (i.e., proteins with three homologous chains in the asymmetric unit related by NCS), and 59 nonsymmetrical trimers (i.e., proteins with at least three different chains in the asymmetric unit). The corresponding query protocols for the PDB are listed in Table 2.2. For our benchmark, we selected only those structures that have a good contact between individual proteins. We define a good contact as a contact with the interface area comprising at least 8% of the accessible solvent area (ASA) of the biggest partner. We required all three protein-protein interfaces within a trimer to satisfy this condition. We retrieved the interface area and the ASA from PDBePISA server [74]. For symmetrical proteins, we computed contacts between different threefold

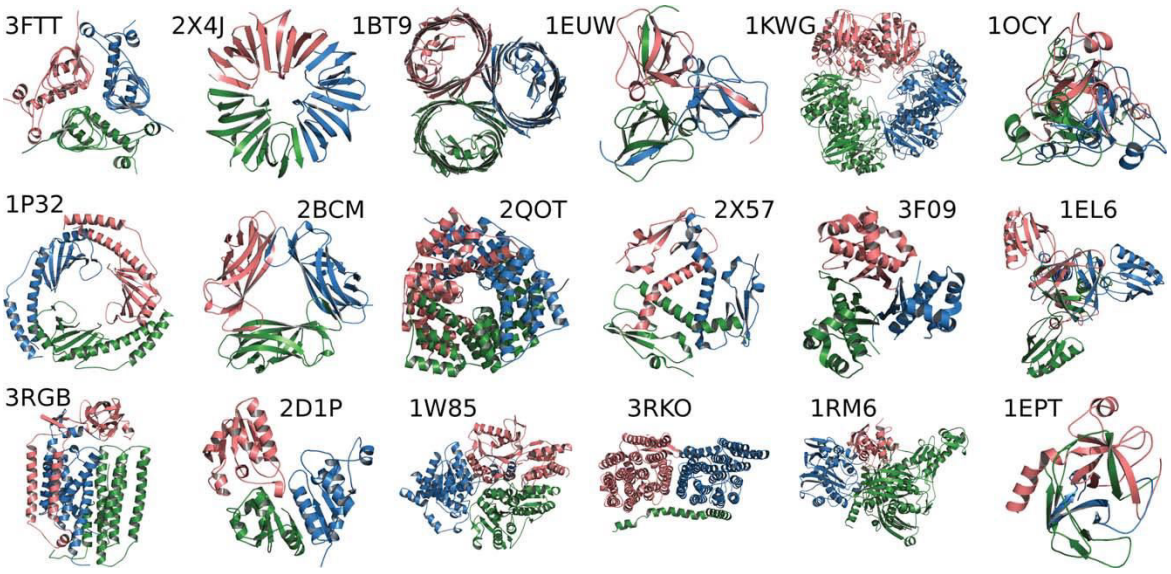


Figure 2.2 Some examples of bound complexes from our trimer benchmark set. Top row: symmetrical structures. Middle row: NCS structures. Bottom row: nonsymmetrical structures. All images were generated using PyMOL [124].

symmetry-mates. Otherwise, we computed contacts between individual proteins. For the remaining symmetrical proteins, we generated trimers according to the BIOMT transform from the PDB file. Any such proteins that lacked a BIOMT record in their PDB file were discarded.

Many structures in the PDB contain three homologous chains with no crystallographic symmetry but which nonetheless possess threefold symmetry. We classify these structures as NCS. In our benchmark set, among all the proteins with three chains in the asymmetric unit, three structures (1F6F, 1U7F, and 2HY5) contain nonhomologues chains without threefold symmetry. We thus classified them as nonsymmetric. Because the PDB contains only a few nonsymmetrical structures with exactly three chains in the asymmetric unit, we also scanned the PDB for higher-order oligomers. We then selected any such structures having triangular pair-wise contacts, and we added the corresponding trimers to the benchmark set.

Table 2.6 lists the proteins selected for the benchmark. This table also lists the IDs of the included chains for nonsymmetrical complexes. To assess the quality of predictions of different docking algorithms objectively, the orientations of all NCS and nonsymmetrical proteins in the benchmark set were randomized before performing any docking calculations.

Generally, it is important to validate a new docking algorithm using proteins whose structures have been determined in the unbound state. Therefore, we looked for crystallographically solved structures which are homologous to the proteins in the bound benchmark,

Table 2.2 Query protocols for the PDB that were used to compose the bound benchmark set.

| Symmetrical | NCS[a] | Non-Symmetrical |
|---|---|---|
| space group: R3 or H3 or P 3 2 1 | number of chains = 3 | number of chains $\geq 3$ |
| X-ray resolution: 0 – 3 Å | X-ray resolution: 0 – 3 Å | X-ray resolution: 0 – 3 Å |
| exclude homology at 70% | exclude homology at 70% | exclude homology at 70% |
| contains only proteins | contains only proteins | contains only proteins |
| | molecular weight $\leq 90$ kDa | number of protein entities $\geq 3$ |
| result count[b]: 85 | result count[b]: 76 | result count[b]: 59 |

[a] Three protein structures, 1F6F, 1U7F, and 2HY5 retrieved by this protocol are non-symmetrical.
[b] After processing of contacts.

but which have different contacts between proteins. We retrieved the PDB codes of homologues proteins from the PDB database (ftp://resources.rcsb.org/sequence/clusters) where all protein chains in the PDB are clustered by BLASTClust algorithm. We used a 95% homology threshold to discriminate between homologous proteins. We then inspected the space group and contacts of all the candidate proteins and selected just seven complexes which are listed in Table 2.3. Among the selected proteins for this unbound benchmark set, four structures possess threefold crystallographic symmetry. These have PDB codes 1F7O, 1IQA, 2E7A, and 3GQH.

Table 2.3 The complexes of the unbound trimer benchmark set.

| Bound complex[a] | Unbound components[b] (space group[c]) | | |
|---|---|---|---|
| 1A3F | 1POB:A (C 2 2 21) | | |
| 1F7O | 1DUT:A (P 63) | 1DUT:B (P 63) | 1F7R:A (P 21 3) |
| 1IQA | 3ME2:A (P 63) | 1QBQ:A (P 61) | 3QBQ:C (P 3) |
| 1U7F | 1DEV:A (P 31 2 1) | 1U7F:B (P 21 21 21) | 1DEV:C (P 31 2 1) |
| 2R3U | 1TE0:A (I 2 3) | | |
| 2E7A | 3L9J:T (P 63 2 2) | | |
| 3GQH | 3GQK:A (P 21 3) | | |

[a]The PDB IDs of trimers from the bound benchmark.
[b]The PDB IDs and chain IDs of monomers in the unbound state.
[c]The space group of each monomer.

### 2.2.4 Comparison with SymmDock, CombDock, and HADDOCK

We compared the performance of the DockTrina algorithm with the results obtained from SymmDock [123] for the bound symmetric benchmark trimers and with results from CombDock [54] for the bound NCS and nonsymmetric benchmark trimers. We also predicted structures of the first 20 nonsymmetrical complexes from our benchmark set using the HADDOCK algorithm [60]. We used default settings provided by SymmDock, CombDock, and HADDOCK. For HADDOCK, we additionally enforced contacts between the chains to favor triangular trimers in the predictions. As the input for SymmDock, we used structures of monomers with $C_3$ symmetry. For CombDock and HADDOCK, we used the structures of protein trimers as input. The running time of these algorithms strongly depends on the size of a protein complex. Nonetheless, SymmDock and CombDock algorithms are very fast. For the smallest and the largest nonsymmetrical trimers, CombDock needs about 15 seconds and 5 minutes, respectively. The corresponding running times for SymmDock on symmetrical complexes are 4 seconds and 1 minute, respectively. HADDOCK requires more time to obtain results for a particular trimer. It needs 1.5 minutes and 25 minutes to obtain 100 predictions for the smallest and the largest nonsymmetrical trimers, respectively. We did not compare DockTrina against Multi-LZerD or ATTRACT/PTools due to their very high execution times (typically 3 hours or more per complex).

## 2.3 Results and Discussion

### 2.3.1 Bound Trimer Assembly Results

We first tested the DockTrina algorithm on our benchmark set of 220 bound protein complexes. Table 2.4 summarizes the performance of DockTrina on this benchmark set obtained when using 1,000 pair-wise contacts between the monomers generated by Hex. Table 2.6 lists the detailed results for this benchmark, which is split into three classes corresponding to symmetrical, NCS, and nonsymmetrical protein complexes. To quantify in a simple way the quality of DockTrina's predictions, we use a numerical quality measure for a protein trimer. We say that the quality-one corresponds to a trimer prediction with all three pair-wise RMSDs smaller than 3.0 Å, quality-two corresponds to a prediction with all three pair-wise RMSDs smaller than 5.0 Å, and quality-three corresponds to a prediction with all three pair-wise RMSDs smaller than 10.0 Å. A pair-wise RMSD is determined by superposing the receptor protein from the prediction with the receptor protein from the reference complex and computing the all-atom RMSD between the ligand proteins. These criteria are summarized in Table 2.5. We also characterize each prediction using the three ranks $i$, $j$,

and $k$ of pair-wise contacts provided by Hex. Given three monomers A, B, C in a trimer, these ranks correspond to the ranks calculated by Hex for the pair-wise protein contacts, A–B, B–C, and A–C. Finally, we also score and rank the complete trimer using our own scoring function (see Eq. (2.2)). For the symmetrical and NCS cases, we successfully

Table 2.4 Summary of results for the bound and unbound benchmarks.

| Classification | Quality = 1/2/3 | | Quality = 1 | |
|---|---|---|---|---|
| | Top1[a] | Top10[a] | Top1[a] | Top10[a] |
| Bound benchmark: | | | | |
| Symmetrical (85[b]) | 58/70 | 64/70 | 41/65 | 54/65 |
| NCS[c] (76[b]) | 31/69 | 55/69 | 18/64 | 43/64 |
| Non-symmetrical (61[b]) | 27/31 | 27/31 | 20/25 | 25/25 |
| Total bound (220[b]) | 116/170 | 146/170 | 79/154 | 122/154 |
| Unbound benchmark: | | | | |
| Total unbound (7[b]) | 0/5 | 4/5 | 0/4 | 2/4 |

[a]The numbers $x/y$ represent the number of correctly predicted trimers ($x$), and the number of structures with at least one feasible pair-wise contact for all 3 pairs from Hex ($y$).
[b]The total number of structures in this class.
[c]Non-crystallographic symmetry.

Table 2.5 Trimer docking quality criteria.

| Quality | RMSD$^{AB}$, Å | | RMSD$^{BC}$, Å | | RMSD$^{CA}$, Å |
|---|---|---|---|---|---|
| 1 | $0 < x \leq 3$ | & | $0 < x \leq 3$ | & | $0 < x \leq 3$ |
| 2 | $3 < x \leq 5$ | & | $3 < x \leq 5$ | & | $3 < x \leq 5$ |
| 3 | $5 < x \leq 10$ | & | $5 < x \leq 10$ | & | $5 < x \leq 10$ |

Here, $x$ stands for the pair-wise RMSD between two monomers.

predicted near-native assemblies (quality-one) for 59 of the 161 trimers; and in 97 cases, we predicted near-native assemblies within the top ten models. We should mention that we did not use any symmetry information for these two classes of complex. The results for the NCS complexes are slightly worse than for the symmetrical cases because we treat the three monomers of these complexes as nonidentical, and thus we discriminate between A–B–C and A–C–B complexes, for example. If these complexes have threefold symmetry, the trimers A–B–C and A–C–B are theoretically equivalent. However, the calculated RMSD between the A–B contact in the A–B–C complex and in the A–C–B complex may be quite large. Consequently, near-native A–C–B predictions in NCS complexes can be classified as

incorrect, even though they are often ranked very highly. For symmetrical complexes, we avoid this problem by using the BIOMT matrix to transform each A–C–B prediction into the A–B–C frame.

From the group of 59 nonsymmetrical trimers, we obtained near-native (quality-one) predictions for 20 complexes ranked first, and 25 complexes within the top ten predictions. Such predictions could subsequently be used to help predict the structures of higher-order oligomers. However, the algorithm often fails when docking very large structures because there is a limitation on the size of the monomers that can be docked using Hex. This occurs for 40 of the nonsymmetrical trimers in our benchmark set.

Another structural feature which can influence successful assembly is the size of the protein-protein interface. For example, the complexes 1EPT, 1LWU, and 1M93 contain large subunits, nonetheless correct predictions were ranked first in all three cases due to their large interfacial areas. The importance of the size of interfacial area can be seen even more clearly with the complex 2WNV, which is a dimer of nonsymmetrical trimers. In this case, DockTrina finds as its first solution the correct assembly of monomers with an average pair-wise interface area of 934.5 $\mathring{A}^2$. However, DockTrina fails on another assembly of monomers from the same structure, where the correct crystallographic solution has an average interface area of only 617.4 $\mathring{A}^2$ (see Table 2.6 ).

Overall, for 220 complexes from our benchmark, the DockTrina algorithm successfully ranked first 116 acceptable predictions (quality-one, -two, or -three) and 146 acceptable predictions were ranked in the top ten predictions. Given that Hex did not produce any useful pair-wise contacts for 50 cases, even if DockTrina performs perfectly we can only expect to obtain a maximum of 170 correct predictions out of 220. For near-native predictions (quality-one), we obtained 122 trimers ranked in the top ten predictions, and 79 trimers ranked first. Hence, the overall results from DockTrina are impressive.

Figure 2.2 shows some examples of the complexes from the bound benchmark set. We present both cases, where DockTrina produces correct and incorrect predictions. For example, DockTrina obtains correct predictions for 1W85, 1RM6, 1EPT nonsymmetrical protein complexes from the figure. It also correctly predicts symmetrical 3FTT, 1EUW, 1OCY, and NCS 2Q0T, 1EL6 complexes. In several cases, DockTrina obtained a much better final rank compared to the three individual ranks of protein contacts provided by Hex. For example, for 1M1J, the final rank is one, whereas the best individual contact rank is 86 (see Table 2.6).

Table 2.6 Bound benchmark set for the DockTrina algorithm.

| PDB ID[a] | SymmN/ N chains[b] | Space Group/ Chain IDs[c] | Quality = 1,2,3 | | | | | | Quality = 1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Quality[d] | Rank[e] | RMSD[f] | i[g] | j[h] | k[i] | Quality[d] | Rank[e] | RMSD[f] | i[g] | j[h] | k[i] |
| Symmetrical Complexes (85) | | | | | | | | | | | | | | |
| 1ALY | 3 | H 3 | 1 | 1 | 1.74 | 9 | 1 | 1 | 1 | 1 | 1.74 | 9 | 1 | 1 |
| 1BT9 | 3 | P 3 2 1 | 2 | 1 | 2.87 | 2 | 2 | 2 | – | – | – | – | – | – |
| 1CB0 | 3 | P 3 2 1 | 2 | 1 | 8.00 | 1 | 3 | 1 | 1 | 21 | 7.99 | 1 | 3 | 360 |
| 1COI | 6 | P 3 2 1 | 2 | 1 | 0.59 | 387 | 4 | 438 | 1 | 5 | 0.69 | 567 | 973 | 2 |
| 1DCS | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1DF4 | 3 | H 3 | 1 | 1 | 0.52 | 67 | 652 | 108 | 1 | 1 | 0.52 | 67 | 652 | 108 |
| 1EUW | 3 | H 3 | 1 | 1 | 3.14 | 1 | 1 | 1 | 1 | 1 | 3.14 | 1 | 1 | 1 |
| 1F7L | 3 | H 3 | 2 | 3 | 1.74 | 61 | 83 | 211 | 1 | 173 | 2.93 | 61 | 83 | 401 |
| 1FNJ | 3 | H 3 | 1 | 1 | 1.88 | 1 | 1 | 1 | 1 | 1 | 1.88 | 1 | 1 | 1 |
| 1H9J | 3 | P 3 2 1 | 1 | 1 | 1.93 | 3 | 1 | 2 | 1 | 1 | 1.93 | 3 | 1 | 2 |
| 1HTN | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1HUP | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1IHC | 3 | H 3 | 1 | 1 | 1.86 | 1 | 3 | 2 | 1 | 1 | 1.86 | 1 | 3 | 2 |
| 1KFN | 3 | H 3 | 1 | 1 | 0.55 | 982 | 259 | 384 | 1 | 1 | 0.55 | 982 | 259 | 384 |
| 1KWG | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1MG1 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1MZY | 3 | H 3 | 2 | 1 | 2.63 | 1 | 2 | 2 | 1 | 2 | 3.17 | 870 | 1 | 1 |
| 1OCY | 3 | P 3 2 1 | 1 | 1 | 1.54 | 1 | 2 | 1 | 1 | 1 | 1.54 | 1 | 2 | 1 |
| 1OK8 | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1PHO | 3 | P 3 2 1 | 2 | 1 | 3.05 | 4 | 2 | 1 | 1 | 5 | 3.77 | 1 | 3 | 8 |
| 1PIQ | 3 | P 3 2 1 | 2 | 3 | 1.14 | 92 | 1 | 240 | 1 | 62 | 1.32 | 14 | 23 | 138 |
| 1QHV | 3 | P 3 2 1 | 1 | 1 | 2.34 | 1 | 1 | 2 | 1 | 1 | 2.34 | 1 | 1 | 2 |
| 1TD4 | 3 | H 3 | 1 | 95 | 1.52 | 306 | 451 | 154 | 1 | 95 | 1.52 | 306 | 451 | 154 |
| 1UKU | 3 | P 3 2 1 | 2 | 1 | 1.61 | 1 | 7 | 2 | 1 | 2 | 4.46 | 2 | 2 | 1 |
| 1VMH | 3 | H 3 | 1 | 1 | 3.63 | 1 | 1 | 1 | 1 | 1 | 3.63 | 1 | 1 | 1 |
| 1WA0 | 3 | H 3 | 1 | 1 | 5.92 | 1 | 1 | 1 | 1 | 1 | 5.92 | 1 | 1 | 1 |
| 1WM3 | 1 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1XQE | 3 | H 3 | 1 | 1 | 4.24 | 1 | 1 | 1 | 1 | 1 | 4.24 | 1 | 1 | 1 |
| 1YC9 | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1YQ8 | 3 | H 3 | 1 | 1 | 1.85 | 1 | 1 | 4 | 1 | 1 | 1.85 | 1 | 1 | 4 |
| 2B2H | 3 | H 3 | 1 | 1 | 2.81 | 1 | 1 | 1 | 1 | 1 | 2.81 | 1 | 1 | 1 |
| 2BSF | 3 | P 3 2 1 | 1 | 1 | 4.10 | 2 | 33 | 3 | 1 | 1 | 4.10 | 2 | 33 | 3 |
| 2CV6 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2CW4 | 3 | H 3 | 1 | 6 | 2.25 | 1 | 8 | 4 | 1 | 6 | 2.25 | 1 | 8 | 4 |
| 2FB6 | 3 | H 3 | 1 | 12 | 1.84 | 22 | 8 | 291 | 1 | 12 | 1.84 | 22 | 8 | 291 |
| 2FKK | 3 | P 3 2 1 | 1 | 1 | 2.21 | 3 | 1 | 1 | 1 | 1 | 2.21 | 3 | 1 | 1 |
| 2NMU | 3 | P 3 2 1 | 1 | 1 | 1.94 | 3 | 1 | 1 | 1 | 1 | 1.94 | 3 | 1 | 1 |
| 2NZ6 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2POR | 3 | H 3 | 2 | 1 | 4.82 | 2 | 1 | 1 | 1 | 2 | 1.63 | 1 | 5 | 6 |
| 2Q2X | 3 | P 3 2 1 | 3 | 1 | 8.44 | 1 | 27 | 18 | – | – | – | – | – | – |
| 2QLK | 3 | H 3 | 2 | 1 | 1.52 | 357 | 179 | 25 | 1 | 16 | 5.60 | 12 | 179 | 25 |
| 2R32 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2RFR | 3 | P 3 2 1 | 1 | 1 | 1.48 | 2 | 2 | 1 | 1 | 1 | 1.48 | 2 | 2 | 1 |
| 2STD | 3 | P 3 2 1 | 3 | 1 | 9.73 | 1 | 1 | 3 | 1 | 45 | 3.41 | 6 | 35 | 49 |
| 2VBK | 1 | H 3 | 3 | 1 | 9.42 | 3 | 171 | 1 | – | – | – | – | – | – |
| 2VJI | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2VNL | 3 | H 3 | 1 | 1 | 2.19 | 2 | 1 | 1 | 1 | 1 | 2.19 | 2 | 1 | 1 |
| 2WPY | 3 | P 3 2 1 | 2 | 1 | 0.80 | 35 | 271 | 24 | 1 | 3 | 0.82 | 360 | 404 | 4 |
| 2WR8 | 3 | H 3 | 1 | 1 | 1.98 | 1 | 11 | 1 | 1 | 1 | 1.98 | 1 | 11 | 1 |
| 2X4J | 3 | H 3 | 1 | 288 | 5.64 | 105 | 271 | 14 | 1 | 288 | 5.64 | 105 | 271 | 14 |
| 2XQH | 3 | H 3 | 2 | 1 | 1.42 | 4 | 2 | 15 | 1 | 243 | 6.02 | 8 | 122 | 30 |
| 2XZR | 3 | P 3 2 1 | 2 | 1 | 0.53 | 36 | 861 | 265 | 1 | 2 | 0.61 | 740 | 234 | 61 |
| 3B9W | 3 | H 3 | 1 | 1 | 3.16 | 1 | 1 | 1 | 1 | 1 | 3.16 | 1 | 1 | 1 |
| 3BZQ | 3 | H 3 | 1 | 1 | 2.88 | 1 | 1 | 1 | 1 | 1 | 2.88 | 1 | 1 | 1 |
| 3CI3 | 3 | H 3 | 1 | 1 | 6.00 | 1 | 1 | 1 | 1 | 1 | 6.00 | 1 | 1 | 1 |
| 3DJ4 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 3EG4 | 3 | H 3 | 1 | 1 | 2.00 | 4 | 2 | 1 | 1 | 1 | 2.00 | 4 | 2 | 1 |
| 3FTT | 3 | H 3 | 1 | 1 | 4.54 | 53 | 2 | 1 | 1 | 1 | 4.54 | 53 | 2 | 1 |
| 3FWU | 3 | H 3 | 1 | 1 | 3.00 | 1 | 1 | 2 | 1 | 1 | 3.00 | 1 | 1 | 2 |
| 3GVL | 3 | H 3 | 1 | 1 | 4.27 | 1 | 2 | 2 | 1 | 1 | 4.27 | 1 | 2 | 2 |
| 3GWM | 3 | H 3 | 3 | 109 | 3.74 | 461 | 94 | 54 | – | – | – | – | – | – |
| 3H56 | 3 | H 3 | 1 | 1 | 3.64 | 2 | 1 | 5 | 1 | 1 | 3.64 | 2 | 1 | 5 |
| 3HWU | 3 | H 3 | 1 | 2 | 1.98 | 1 | 1 | 1 | 1 | 2 | 1.98 | 1 | 1 | 1 |
| 3I87 | 3 | P 3 2 1 | 1 | 1 | 3.75 | 6 | 1 | 1 | 1 | 1 | 3.75 | 6 | 1 | 1 |
| 3IJ4 | 3 | H 3 | 1 | 1 | 4.76 | 3 | 1 | 1 | 1 | 1 | 4.76 | 3 | 1 | 1 |
| 3KWE | 3 | H 3 | 1 | 4 | 3.72 | 10 | 28 | 87 | 1 | 4 | 3.72 | 10 | 28 | 87 |
| 3LAA | 3 | H 3 | 1 | 1 | 0.07 | 284 | 7 | 14 | 1 | 1 | 0.07 | 284 | 7 | 14 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3LGU | 1 | H 3 | 2 | 1207 | 4.01 | 52 | 543 | 489 | – | – | – | – | – | – |
| 3LQW | 3 | H 3 | 1 | 1 | 3.17 | 1 | 2 | 1 | 1 | 1 | 3.17 | 1 | 2 | 1 |
| 3M73 | 3 | H 3 | 1 | 1 | 2.68 | 1 | 1 | 1 | 1 | 1 | 2.68 | 1 | 1 | 1 |
| 3MC3 | 3 | H 3 | 1 | 1 | 1.04 | 2 | 12 | 619 | 1 | 1 | 1.04 | 2 | 12 | 619 |
| 3MDX | 3 | H 3 | 1 | 1 | 2.69 | 1 | 1 | 1 | 1 | 1 | 2.69 | 1 | 1 | 1 |
| 3N4H | 3 | P 3 2 1 | 1 | 1 | 4.03 | 1 | 1 | 1 | 1 | 1 | 4.03 | 1 | 1 | 1 |
| 3NUM | 3 | H 3 | 2 | 4 | 4.70 | 35 | 5 | 34 | 1 | 60 | 6.14 | 25 | 29 | 96 |
| 3OC7 | 3 | H 3 | 1 | 1 | 4.78 | 1 | 1 | 2 | 1 | 1 | 4.78 | 1 | 1 | 2 |
| 3PMO | 3 | H 3 | 1 | 1 | 3.03 | 1 | 1 | 2 | 1 | 1 | 3.03 | 1 | 1 | 2 |
| 3PRN | 3 | H 3 | 2 | 1 | 5.64 | 4 | 1 | 1 | 1 | 7 | 4.91 | 1 | 4 | 31 |
| 3Q1X | 3 | H 3 | 2 | 1 | 5.82 | 1 | 1 | 1 | 1 | 2 | 6.30 | 1 | 1 | 2 |
| 3QR8 | 3 | P 3 2 1 | 1 | 1 | 2.13 | 1 | 2 | 1 | 1 | 1 | 2.13 | 1 | 2 | 1 |
| 3QUW | 3 | H 3 | 2 | 1 | 1.87 | 6 | 3 | 1 | 1 | 2 | 4.01 | 1 | 2 | 1 |
| 3R3R | 3 | H 3 | 1 | 13 | 2.71 | 91 | 1 | 6 | 1 | 13 | 2.71 | 91 | 1 | 6 |
| 3TDT | 3 | H 3 | 1 | 1 | 3.62 | 1 | 1 | 1 | 1 | 1 | 3.62 | 1 | 1 | 1 |
| 3TG7 | 3 | H 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 4AC3 | 3 | P 3 2 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 4TSV | 3 | H 3 | 1 | 1 | 0.68 | 10 | 3 | 488 | 1 | 1 | 0.68 | 10 | 3 | 488 |
| | | | Top1 58/85/70[j] | | | Top10 64/85/70[j] | | | Top1 41/85/65[j] | | | Top10 54/85/65[j] | | |

NCS Complexes (76)

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1A3F | 3 | P 21 21 21 | 3 | 83171 | 9.49 | 204 | 180 | 793 | 1 | 90648 | 3.67 | 749 | 846 | 793 |
| 1C28 | 3 | P 61 | 1 | 1 | 1.72 | 3 | 1 | 1 | 1 | 1 | 1.72 | 3 | 1 | 1 |
| 1COS | 1 | P 21 21 21 | 1 | 12 | 0.81 | 165 | 2 | 147 | 1 | 12 | 0.81 | 165 | 2 | 147 |
| 1EL6 | 3 | P 21 21 21 | 1 | 1 | 2.22 | 4 | 1 | 1 | 1 | 1 | 2.22 | 4 | 1 | 1 |
| 1F7O | 3 | P 21 21 21 | 1 | 1 | 1.61 | 1 | 1 | 2 | 1 | 1 | 1.61 | 1 | 1 | 2 |
| 1FTF | 3 | C 1 2 1 | 2 | 6 | 2.89 | 25 | 1 | 59 | 1 | 77 | 4.01 | 299 | 25 | 1 |
| 1GCM | 3 | P 1 21 1 | 3 | 14 | 0.86 | 1 | 657 | 297 | 1 | 45 | 0.97 | 136 | 11 | 219 |
| 1IDP | 3 | P 1 21 1 | 2 | 1 | 3.72 | 1 | 2 | 2 | 1 | 7 | 3.72 | 1 | 2 | 2 |
| 1IQA | 3 | P 21 21 21 | 1 | 1 | 2.96 | 1 | 2 | 2 | 1 | 1 | 2.96 | 1 | 2 | 2 |
| 1JCD | 3 | P 1 | 2 | 12 | 0.38 | 16 | 29 | 856 | 1 | 168 | 0.54 | 558 | 765 | 374 |
| 1KK6 | 3 | P 21 21 21 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1KKE | 1 | P 21 21 21 | 2 | 343 | 6.39 | 15 | 113 | 2 | 1 | 1863 | 5.78 | 15 | 113 | 11 |
| 1LW1 | 2 | C 1 2 1 | 1 | 1 | 2.82 | 1 | 2 | 1 | 1 | 1 | 2.82 | 1 | 2 | 1 |
| 1O8O | 3 | I 41 2 2 | 1 | 6 | 3.44 | 2 | 7 | 1 | 1 | 6 | 3.44 | 2 | 7 | 1 |
| 1ODE | 3 | P 21 21 21 | 3 | 1 | 1.85 | 9 | 2 | 6 | 1 | 248 | 2.96 | 4 | 6 | 5 |
| 1P32 | 3 | P 1 21 1 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1Q5U | 3 | P 21 21 2 | 2 | 20 | 1.86 | 1 | 7 | 3 | 1 | 23 | 2.24 | 2 | 2 | 3 |
| 1QU9 | 3 | P 2 2 2 | 1 | 4 | 2.16 | 2 | 1 | 2 | 1 | 4 | 2.16 | 2 | 1 | 2 |
| 1RGX | 6 | C 1 2 1 | 1 | 4 | 1.93 | 2 | 3 | 1 | 1 | 4 | 1.93 | 2 | 3 | 1 |
| 1S2L | 6 | P 43 21 2 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1TD3 | 3 | C 1 2 1 | 1 | 3054 | 3.16 | 132 | 33 | 687 | 1 | 3054 | 3.16 | 132 | 33 | 687 |
| 1TDT | 3 | P 1 21 1 | 1 | 1 | 4.80 | 1 | 1 | 1 | 1 | 1 | 4.80 | 1 | 1 | 1 |
| 1U5Y | 3 | C 1 2 1 | 1 | 29 | 2.85 | 28 | 1 | 10 | 1 | 29 | 2.85 | 28 | 1 | 10 |
| 1UDE | 6 | P 21 21 2 | 2 | 1 | 3.70 | 29 | 57 | 18 | 1 | 38 | 3.62 | 48 | 87 | 13 |
| 1VFJ | 3 | P 21 21 21 | 1 | 1 | 2.58 | 1 | 1 | 1 | 1 | 1 | 2.58 | 1 | 1 | 1 |
| 1WP8 | 3 | P 1 | 1 | 4 | 1.24 | 13 | 1 | 2 | 1 | 4 | 1.24 | 13 | 1 | 2 |
| 1WT6 | 3 | P 21 21 21 | 2 | 2 | 0.59 | 903 | 323 | 76 | 1 | 5695 | 1.75 | 8 | 323 | 584 |
| 1WVT | 3 | P 41 21 2 | 1 | 5 | 1.56 | 8 | 14 | 14 | 1 | 5 | 1.56 | 8 | 14 | 14 |
| 1YQ6 | 3 | P 41 21 2 | 1 | 1 | 1.46 | 4 | 1 | 5 | 1 | 1 | 1.46 | 4 | 1 | 5 |
| 1ZVB | 3 | C 1 2 1 | 1 | 9 | 0.95 | 698 | 25 | 4 | 1 | 9 | 0.95 | 698 | 25 | 4 |
| 2BA2 | 1 | P 1 21 1 | 3 | 1 | 0.36 | 916 | 156 | 111 | 1 | 4 | 0.51 | 325 | 25 | 598 |
| 2BAZ | 3 | P 43 | 1 | 2 | 2.23 | 1 | 2 | 3 | 1 | 2 | 2.23 | 1 | 2 | 3 |
| 2BCM | 3 | P 41 | 1 | 862 | 3.70 | 232 | 104 | 35 | 1 | 862 | 3.70 | 232 | 104 | 35 |
| 2BSD | 3 | P 21 21 21 | 2 | 1 | 1.98 | 1 | 2 | 2 | 1 | 3 | 2.10 | 2 | 6 | 2 |
| 2CHC | 3 | C 2 2 21 | 1 | 6 | 2.04 | 4 | 2 | 1 | 1 | 6 | 2.04 | 4 | 2 | 1 |
| 2CU5 | 3 | C 1 2 1 | 3 | 1 | 2.77 | 1 | 2 | 3 | 1 | 24 | 2.77 | 1 | 2 | 3 |
| 2DJ6 | 3 | P 21 21 21 | 1 | 1 | 2.12 | 1 | 3 | 5 | 1 | 1 | 2.12 | 1 | 3 | 5 |
| 2E2A | 3 | P 41 21 2 | 1 | 1 | 1.71 | 1 | 1 | 2 | 1 | 1 | 1.71 | 1 | 1 | 2 |
| 2E7A | 3 | P 21 21 21 | 1 | 6 | 0.76 | 225 | 3 | 20 | 1 | 6 | 0.76 | 225 | 3 | 20 |
| 2FVH | 3 | P 6 | 3 | 65609 | 7.60 | 723 | 984 | 69 | – | – | – | – | – | – |
| 2GDG | 3 | P 63 | 1 | 2 | 2.45 | 1 | 1 | 1 | 1 | 2 | 2.45 | 1 | 1 | 1 |
| 2GTR | 3 | C 1 2 1 | 1 | 4 | 2.24 | 28 | 21 | 58 | 1 | 4 | 2.24 | 28 | 21 | 58 |
| 2I9D | 3 | P 61 | 1 | 5 | 2.84 | 92 | 1 | 29 | 1 | 5 | 2.84 | 92 | 1 | 29 |
| 2IC7 | 3 | C 1 2 1 | 1 | 1 | 1.79 | 2 | 1 | 14 | 1 | 1 | 1.79 | 2 | 1 | 14 |
| 2IG8 | 3 | P 41 2 2 | 1 | 1 | 1.86 | 1 | 1 | 1 | 1 | 1 | 1.86 | 1 | 1 | 1 |
| 2IUM | 3 | C 1 2 1 | 3 | 1 | 5.31 | 45 | 6 | 8 | 1 | 4 | 3.30 | 8 | 259 | 6 |
| 2PBQ | 3 | P 1 21 1 | 1 | 326 | 2.89 | 48 | 950 | 52 | 1 | 326 | 2.89 | 48 | 950 | 52 |
| 2Q0T | 3 | P 1 21 1 | 3 | 4 | 2.88 | 2 | 1 | 2 | 1 | 12 | 2.88 | 2 | 1 | 2 |
| 2Q4I | 3 | P 21 21 21 | 1 | 1 | 0.97 | 3 | 634 | 2 | 1 | 1 | 0.97 | 3 | 634 | 2 |
| 2R3U | 3 | C 1 2 1 | 1 | 7 | 3.58 | 14 | 9 | 3 | 1 | 7 | 3.58 | 14 | 9 | 3 |
| 2WX3 | 1 | P 32 2 1 | 2 | 1 | 0.36 | 937 | 292 | 208 | 1 | 4 | 0.41 | 898 | 283 | 276 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2X57 | 6 | P 41 21 2 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2YW6 | 12 | P 62 2 2 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2ZFC | 3 | H 3 | 3 | 1 | 0.15 | 916 | 493 | 367 | 1 | 9 | 0.66 | 4 | 230 | 302 |
| 2ZHY | 3 | C 2 2 21 | 3 | 14 | 6.99 | 1 | 2 | 176 | – | – | – | – | – | – |
| 3AA8 | 3 | P 61 | 3 | 1 | 2.51 | 2 | 2 | 1 | 1 | 14 | 2.51 | 2 | 2 | 1 |
| 3B93 | 3 | C 1 2 1 | 3 | 82 | 7.47 | 11 | 631 | 1 | – | – | – | – | – | – |
| 3BHP | 3 | C 1 2 1 | 3 | 59765 | 6.97 | 3 | 261 | 223 | – | – | – | – | – | – |
| 3CM1 | 1 | P 41 | 1 | 5 | 3.07 | 46 | 42 | 27 | 1 | 5 | 3.07 | 46 | 42 | 27 |
| 3D9X | 3 | P 1 | 1 | 1 | 0.31 | 171 | 57 | 15 | 1 | 1 | 0.31 | 171 | 57 | 15 |
| 3DA0 | 3 | P 1 21 1 | 1 | 1 | 1.57 | 2 | 1 | 2 | 1 | 1 | 1.57 | 2 | 1 | 2 |
| 3DLI | 3 | P 21 21 21 | 2 | 1 | 7.18 | 11 | 112 | 5 | – | – | – | – | – | – |
| 3EMF | 3 | I 2 2 2 | 1 | 4 | 1.57 | 1 | 2 | 2 | 1 | 4 | 1.57 | 1 | 2 | 2 |
| 3EMO | 3 | C 1 2 1 | 1 | 1 | 0.60 | 11 | 198 | 310 | 1 | 1 | 0.60 | 11 | 198 | 310 |
| 3EXW | 3 | P 21 21 2 | 2 | 6 | 4.80 | 10 | 70 | 9 | 1 | 70 | 3.46 | 70 | 9 | 365 |
| 3F09 | 3 | P 21 21 21 | 2 | 5 | 2.79 | 125 | 16 | 1 | 1 | 15 | 3.26 | 362 | 56 | 1 |
| 3FD9 | 3 | P 21 21 2 | – | – | – | – | – | – | – | – | – | – | – | – |
| 3GQH | 3 | P 21 21 21 | 3 | 1 | 3.28 | 5 | 3 | 4 | 1 | 27 | 3.28 | 5 | 3 | 4 |
| 3H6X | 3 | C 1 2 1 | 3 | 1 | 1.84 | 2 | 2 | 15 | 1 | 15 | 1.84 | 2 | 2 | 15 |
| 3HFE | 3 | C 1 2 1 | 1 | 2 | 0.87 | 33 | 2 | 560 | 1 | 2 | 0.87 | 33 | 2 | 560 |
| 3N4G | 3 | C 1 2 1 | 1 | 7 | 3.63 | 1 | 2 | 1 | 1 | 7 | 3.63 | 1 | 2 | 1 |
| 3QXI | 3 | C 1 2 1 | 1 | 2 | 4.04 | 1 | 1 | 1 | 1 | 2 | 4.04 | 1 | 1 | 1 |
| 3R1W | 3 | P 1 21 1 | 1 | 1 | 2.48 | 1 | 2 | 1 | 1 | 1 | 2.48 | 1 | 2 | 1 |
| 3STI | 3 | P 31 | – | – | – | – | – | – | – | – | – | – | – | – |
| 3SWY | 1 | P 1 21 1 | 1 | 1 | 1.28 | 1 | 3 | 1 | 1 | 1 | 1.28 | 1 | 3 | 1 |
| 3T97 | 1 | P 21 21 21 | 1 | 4 | 0.69 | 178 | 837 | 2 | 1 | 4 | 0.69 | 178 | 837 | 2 |
| | | | Top1 31/76/69[j] | | | Top10 55/76/69[j] | | | Top1 18/76/64[j] | | | Top10 43/76/64[j] | | |

Non-Symmetrical Complexes (59)

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1AYM | 1 | 1 2 3 | 1 | 1 | 0.88 | 1 | 1 | 1 | 1 | 1 | 0.88 | 1 | 1 | 1 |
| 1B35 | 60 | A B C | 1 | 1 | 0.99 | 1 | 1 | 2 | 1 | 1 | 0.99 | 1 | 1 | 2 |
| 1BEV | 60 | 1 2 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1DGW | 3 | A X Y | 1 | 1 | 0.48 | 1 | 1 | 5 | 1 | 1 | 0.48 | 1 | 1 | 5 |
| 1E6Y | 2 | C B D | – | – | – | – | – | – | – | – | – | – | – | – |
| 1EPT | 1 | A B C | 1 | 1 | 0.52 | 1 | 1 | 1 | 1 | 1 | 0.52 | 1 | 1 | 1 |
| 1EYS | 1 | M H L | – | – | – | – | – | – | – | – | – | – | – | – |
| 1F6F | 1 | A B C | – | – | – | – | – | – | – | – | – | – | – | – |
| 1FI8 | 2 | C D F | – | – | – | – | – | – | – | – | – | – | – | – |
| 1HBN | 2 | E F D | – | – | – | – | – | – | – | – | – | – | – | – |
| 1HIA | 1 | A B I | 3 | 831 | 7.51 | 65 | 225 | 1 | – | – | – | – | – | – |
| 1J34 | 1 | A B C | 3 | 1 | 8.71 | 1 | 1 | 330 | – | – | – | – | – | – |
| 1L1O | 2 | A B C | 3 | 267 | 9.19 | 11 | 1 | 19 | – | – | – | – | – | – |
| 1LWU | – | A B C | 2 | 1 | 0.36 | 204 | 44 | 784 | 1 | 3 | 0.98 | 2 | 2 | 1 |
| 1M1J | – | A B C | 1 | 1 | 0.32 | 764 | 453 | 86 | 1 | 1 | 0.32 | 764 | 453 | 86 |
| 1M93 | 1 | A B C | 2 | 1 | 0.15 | 324 | 1 | 2 | 1 | 4 | 1.16 | 1 | 32 | 1 |
| 1MTY | 2 | C E H | 3 | 1 | 6.16 | 54 | 1 | 1 | – | – | – | – | – | – |
| 1O7D | 1 | D C B | – | – | – | – | – | – | – | – | – | – | – | – |
| 1PVC | – | 1 2 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1QQP | 60 | 1 2 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 1RM6 | 2 | A B C | 1 | 1 | 0.88 | 1 | 2 | 1 | 1 | 1 | 0.88 | 1 | 2 | 1 |
| 1SR4 | 1 | A B C | 1 | 1 | 0.95 | 1 | 1 | 1 | 1 | 1 | 0.95 | 1 | 1 | 1 |
| 1U7F | 1 | A B C | 1 | 1 | 0.15 | 4 | 4 | 12 | 1 | 1 | 0.15 | 4 | 4 | 12 |
| 1UNB | 3 | v 3 u | – | – | – | – | – | – | – | – | – | – | – | – |
| 1W85 | 1 | F H G | 1 | 1 | 0.08 | 1 | 1 | 1 | 1 | 1 | 0.08 | 1 | 1 | 1 |
| 2AZE | 2 | A B C | 2 | 1 | 0.67 | 38 | 934 | 124 | 1 | 4 | 1.06 | 1 | 11 | 463 |
| 2D1P | 2 | I E G | – | – | – | – | – | – | – | – | – | – | – | – |
| 2DSR | 1 | I G B | 1 | 1 | 0.57 | 6 | 2 | 395 | 1 | 1 | 0.57 | 6 | 2 | 395 |
| 2E1M | 2 | A B C | – | – | – | – | – | – | – | – | – | – | – | – |
| 2E74 | 2 | B C H | 1 | 1 | 0.43 | 1 | 31 | 2 | 1 | 1 | 0.43 | 1 | 31 | 2 |
| 2F66 | 2 | D E B | 1 | 1 | 2.59 | 3 | 102 | 1 | 1 | 1 | 2.59 | 3 | 102 | 1 |
| 2H88 | 1 | Q P O | – | – | – | – | – | – | – | – | – | – | – | – |
| 2HY5 | 2 | A B C | 1 | 1 | 0.78 | 1 | 51 | 2 | 1 | 1 | 0.78 | 1 | 51 | 2 |
| 2HZS | 2 | B C D | – | – | – | – | – | – | – | – | – | – | – | – |
| 2J3W | 1 | A B E | – | – | – | – | – | – | – | – | – | – | – | – |
| 2J8C | 1 | M L H | – | – | – | – | – | – | – | – | – | – | – | – |
| 2MEV | 1 | 1 2 3 | – | – | – | – | – | – | – | – | – | – | – | – |
| 2QFA | 1 | A B C | 1 | 1 | 1.76 | 190 | 1 | 1 | 1 | 1 | 1.76 | 190 | 1 | 1 |
| 2QI9 | 1 | A B F | 1 | 1 | 0.92 | 12 | 1 | 1 | 1 | 1 | 0.92 | 12 | 1 | 1 |
| 2UNB | – | 2 3 j | 1 | 1 | 1.85 | 1 | 6 | 75 | 1 | 1 | 1.85 | 1 | 6 | 75 |
| 2WJN | 1 | L H M | 2 | 1 | 1.20 | 3 | 1 | 5 | 1 | 2 | 2.87 | 3 | 1 | 1 |
| 2WNV | 1 | E D F | 1 | 1 | 0.31 | 1 | 1 | 1 | 1 | 1 | 0.31 | 1 | 1 | 1 |
| 2WNV | 1 | F D C | – | – | – | – | – | – | – | – | – | – | – | – |
| 2WTK | 1 | A B C | – | – | – | – | – | – | – | – | – | – | – | – |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2ZZD | 4 | I G H | 1 | 1 | 1.08 | 1 | 146 | 1 | 1 | 1 | 1.08 | 1 | 146 | 1 |
| 3ARC | – | K Z Y | 3 | 1498885 | 6.37 | 253 | 24 | 732 | – | – | – | – | – | – |
| 3CI0 | 1 | K J I | – | – | – | – | – | – | – | – | – | – | – | – |
| 3CJI | 60 | A B C | – | – | – | – | – | – | – | – | – | – | – | – |
| 3NAP | 60 | A B C | 1 | 1 | 1.97 | 7 | 1 | 23 | 1 | 1 | 1.97 | 7 | 1 | 23 |
| 3P8C | 1 | F D E | 2 | 1 | 0.71 | 1 | 2 | 92 | 1 | 6 | 2.76 | 2 | 1 | 1 |
| 3R0L | 1 | A B D | 3 | 515 | 9.54 | 68 | 1 | 21 | – | – | – | – | – | – |
| 3RGB | 2 | I J K | – | – | – | – | – | – | – | – | – | – | – | – |
| 3RKO | 1 | M N L | – | – | – | – | – | – | – | – | – | – | – | – |
| 3RYC | 1 | E D C | – | – | – | – | – | – | – | – | – | – | – | – |
| 3S6N | 1 | E 2 F | – | – | – | – | – | – | – | – | – | – | – | – |
| 3SQG | 2 | I H G | – | – | – | – | – | – | – | – | – | – | – | – |
| 3U5C | – | C V A | – | – | – | – | – | – | – | – | – | – | – | – |
| 3VBH | 60 | A B C | 1 | 1 | 0.39 | 1 | 1 | 1 | 1 | 1 | 0.39 | 1 | 1 | 1 |
| 4A8X | 1 | A B C | – | – | – | – | – | – | – | – | – | – | – | – |
| | | | Top1 27/59/31[j] | | | Top10 27/59/31[j] | | | Top1 20/59/25[j] | | | Top10 25/59/25[j] | | |

Total (220)

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Top1 116/220/170[j] | | | Top10 146/220/170[j] | | | Top1 79/220/154[j] | | | Top10 122/220/154[j] | | |

[a] Protein ID in the PDB.
[b] Symmetry number as given by PISA [74] for symmetrical or NCS trimers / Number of chains for non-symmetrical trimers.
[c] Space group for symmetrical or NCS trimers / Chain IDs for non-symmetrical trimers.
[d] Quality according to criteria listed in Table 2.5.
[e] Rank provided by our scoring function (see Equation (2.2)).
[f] RMSD between the position of the first monomer $A$ and its transformed position $T^{AA} \cdot A$.
[g] Rank provided by Hex for the first pair of monomers in the trimer.
[h] Rank provided by Hex for the second pair of monomers in the trimer.
[i] Rank provided by Hex for the third pair of monomers in the trimer.
[j] The numbers $x/y/z$ represent the number of correctly predicted trimers ($x$), the total number of trimers in this class ($y$), and the number of structures with at least one feasible pair-wise contact from Hex ($z$).

### 2.3.2 Unbound Trimer Assembly Results

We also tested DockTrina on our benchmark set of seven unbound protein trimers. Here, we measured the RMSD between monomers according to their corresponding bound structures. Table 2.4 gives a summary of the performance of DockTrina on this benchmark. Table 2.7 lists the detailed results of our predictions. We obtained four out of seven acceptable (quality-one, -two, or -three) predictions, and two near-native (quality-one) predictions within the top ten predictions. We should mention that the structures 1F7O, 1IQA, 2E7A, and 3GQH in the unbound benchmark have threefold crystallographic symmetry axes, which probably makes it slightly easier to obtain good predictions in these cases.

Table 2.7 Unbound benchmark set for the DockTrina algorithm.

| PDB ID[1] | Sym[2] | Space Group[3] | Quality = 1,2,3 | | | | | | | Quality = 1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Quality[4] | Rank[5] | B Rank[6] | RMSD[7] | i[8] | j[9] | k[10] | Quality[d] | Rank[e] | RMSD[g] | i[h] | j[i] | k[j] |
| 1A3F | 3 | P 21 21 21 | – | – | 29441 | – | – | – | – | – | – | – | – | – | |
| 1F7O | 3 | P 21 21 21 | 2 | 7 | 1 | 4.64 | 1 | 2 | 2 | 1 | 594 | 1.84 | 2 | 92 | 3 |
| 1IQA | 3 | P 21 21 21 | 2 | 4 | 7 | 4.94 | 2 | 4 | 2 | 1 | 11 | 3.78 | 4 | 3 | 1 |
| 1U7F | 3 | P 21 21 21 | – | – | 186 | – | – | – | – | – | – | – | – | – | – |
| 2E7A | 3 | P 21 21 21 | 1 | 7 | 9 | 5.29 | 2 | 2 | 2 | 1 | 7 | 5.29 | 2 | 2 | 2 |
| 2R3U | 3 | C 1 2 1 | 3 | 1574 | 7 | 3.36 | 73 | 498 | 79 | – | – | – | – | – | – |
| 3GQH | 3 | P 21 21 21 | 1 | 4 | 4 | 3.53 | 8 | 12 | 9 | 1 | 4 | 3.53 | 8 | 12 | 9 |
| Total (7) | | | | | | | | | | | | | | | |
| | | | Top1 0/7/5[11] | | | Top10 4/7/5[k] | | | | Top1 0/7/4[k] | | | Top10 2/7/4[k] | | |

[4]Quality according to criteria listed in Table 2.5.

[5]Rank provided by our scoring function (see Equation (2.2)).

[6]Rank provided by our scoring function for the bound complex, see Table 2.6.

[7]RMSD between the position of the first monomer $A$ and its transformed position $T^{AA} \cdot A$.

[8]Rank provided by Hex for the first pair of monomers in the trimer.

[9]Rank provided by Hex for the second pair of monomers in the trimer.

[10]Rank provided by Hex for the third pair of monomers in the trimer.

[11]The numbers $x/y/z$ represent the number of correctly predicted trimers ($x$), the total number of trimers ($y$), and the number of structures with at least one feasible pair-wise contact from Hex ($z$).

### 2.3.3 Comparison with SymmDock

To assess DockTrina's ability to model symmetrical trimers, we compared DockTrina with the SymmDock algorithm. SymmDock explicitly uses cyclic symmetry information to constrain its exhaustive rigid-body search to a reduced 4D subspace. On the other hand, DockTrina was not specifically developed to model symmetrical structures. It therefore does not use any symmetry information, and instead performs exhaustive rigid-body search in the full 6D space. Hence, we were keen to compare the performance of the two algorithms.

Overall, SymmDock ranked first 65 acceptable predictions (quality-one, -two, or -three) and placed 73 acceptable predictions in the top ten predictions. For near-native (quality-one) predictions, SymmDock ranked 56 trimers within the top ten predictions, and it placed 42 trimers at rank one. These results are slightly better than the DockTrina results, which is perhaps not surprising because DockTrina does not use the $C_3$ symmetry constraint. Nevertheless, the DockTrina results are still highly competitive. For example, DockTrina produced 41 near-native symmetrical trimers at rank one (compared to 42 with SymmDock), and it ranked 54 trimers in the top ten predictions (compared to 56 with SymmDock).

### 2.3.4 Comparison with CombDock

To assess DockTrina's performance on proteins without symmetry, we compared it against the CombDock combinatorial assembly algorithm. However, the success rate of CombDock on our bound benchmark set turned out to be very low. Indeed, CombDock obtained no correct predictions for the 76 NCS proteins, and it produced only two acceptable predictions at rank one for the 59 nonsymmetrical complexes. In contrast, DockTrina ranks a total of 82 NCS and nonsymmetrical trimers of acceptable quality within the top ten models.

We believe the poor performance of CombDock arises because it does not require trimeric contacts in its solutions. More specifically, many of the solutions produced by CombDock are linear trimers with only two contacts between the three monomers. Therefore, for a fair comparison, we excluded linear predictions from the final list of CombDock results. However, it still fails on 135 out of 137 examples used here. On the other hand, DockTrina always requires three contacts between the monomers to build the trimer. Therefore, DockTrina will fail if Hex does not produce at least one acceptable pair-wise contact for each pair of monomers, as for the 1O7D complex, for example (which was one of the two examples correctly predicted by CombDock). Nonetheless, the overall results obtained by DockTrina on these more difficult complexes demonstrate the utility of explicitly searching for triangular contacts.

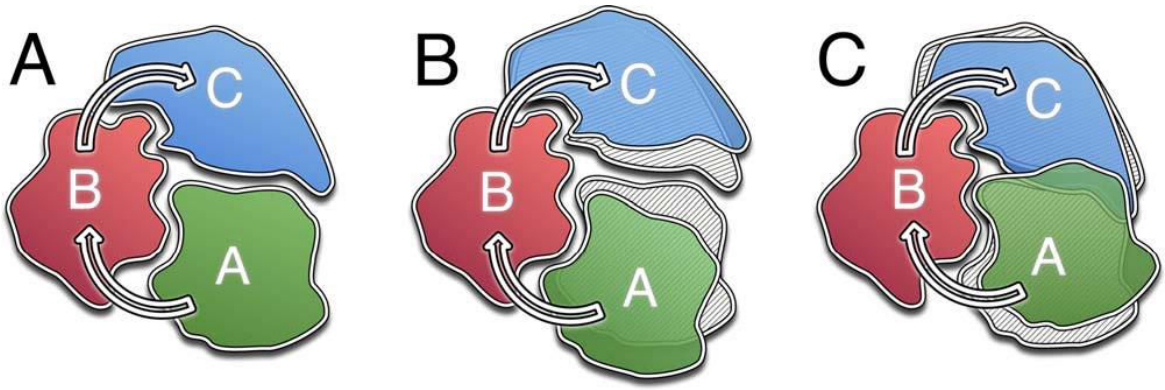


Figure 2.3 (**A**) Native complex of proteins A, B, and C. (**B**) Prediction without a steric clash between proteins. (**C**) Prediction with the same pair-wise RMSD as before but with a steric clash between proteins A and C. DockTrina gives the same geometric penalty term for the trimers in (**B**) and in (**C**).

### 2.3.5 Comparison with HADDOCK

Because HADDOCK was designed to use biological constraints to restrict the search space, it is not well-suited for the blind docking calculations described here. Indeed, HADDOCK requires significant time to prepare the input and calculate even a single complex. Thus, we were not able to test HADDOCK on all of the trimers in our benchmark set. Instead, we ran HADDOCK on a subset of first 20 nonsymmetrical trimers in our benchmark set. However, as it did not find any feasible solutions for this subset, we decided to abandon this comparison due to the manual effort necessary to set up each docking run.

### 2.3.6 The DockTrina Scoring Function

As aforementioned, DockTrina ranks its predictions using a scoring function that combines the pair-wise contact scores from Hex with an empirical geometric penalty term (see Eq. (2.2) ). This is very cheap to evaluate, although it is probably not optimal, as can be seen from Tables 2.6 and 2.7. A more sophisticated scoring function might produce better results. Nonetheless, a crucial advantage of our scoring function is that it detects and penalizes trimers with large A–A′ RMSDs in constant time. We believe that our RMSD-based scoring function produces better predictions compared to more expensive scoring functions that evaluate steric clashes between the monomers, as in CombDock, for example. Figure 2.3 illustrates two predictions ranked equally by DockTrina. However, steric clash-based scoring functions would reject the second prediction, even though it may be very close to the native structure. Although DockTrina predictions may contain clashes, one can straightforwardly refine them using conventional minimization tools.

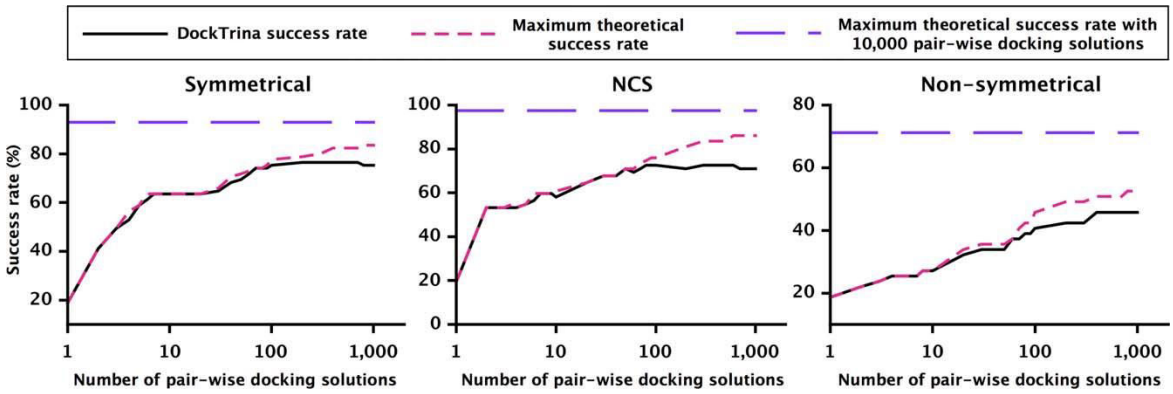


Figure 2.4 Success rates as a function of number of pair-wise docking solutions for symmetrical trimers (left); NCS trimers (middle); and nonsymmetrical trimers (right) from our benchmark. The total number of combinations of the three monomers is given as the third power of the number of pair-wise solutions. Solid black curve represents the DockTrina success rate as a function of number of pair-wise docking solutions. Dashed magenta curve represents the maximum theoretical success rate at a given number of pair-wise docking solutions provided by Hex. Horizontal dashed line represents the maximum theoretical success rate when 10,000 Hex pair-wise docking solutions are considered.

### 2.3.7 DockTrina's Success Rate

To evaluate how DockTrina's success rate depends on the quality of the input pair-wise docking solutions, we analyzed the rigid-body transforms provided by the Hex pair-wise docking algorithm. More precisely, for each pair of monomers, we evaluated the number of acceptable solutions (quality-one, -two, or -three) produced by Hex within the first $M$ pair-wise predictions, where the maximum value of $M$ was 10,000. We consider the maximum theoretical trimer success rate for a given number of pair-wise docking solutions to be the number of trimers with at least one acceptable pair-wise docking solution. Similarly, we calculate the success rate of DockTrina using the number of acceptable trimer predictions found within the top ten DockTrina models for a given number of pair-wise docking solutions. Figure 2.4 shows the success rate of DockTrina (black solid curve) along with the maximum theoretical success rate (magenta dashed curve). The horizontal dashed line represents the maximum theoretical success rate achieved with 10,000 pairwise docking solutions. From 2.4, we can see that DockTrina's success rate increases steadily with the number of input transforms. We can also see that to achieve the greatest trimer assembly performance, it is sufficient to consider only 100 input transforms for symmetrical and NCS complexes. However, around 1000 pair-wise input transforms are needed to maximize trimer assembly performance for nonsymmetrical complexes. Nonetheless, even using the maximum number of pair-wise docking solutions does not give perfect performance due to

the lack of acceptable pair-wise predictions from Hex. Overall, DockTrina's success rate is around 90% of the theoretical maximum, as defined above. This high success rate stems from the ability of DockTrina's scoring function to pull out predictions with low RMSD even if the individual pair-wise docking scores are low (see Eq. (2.2)).

## 2.4 Conclusions

We have presented a new and very efficient algorithm for docking triangular protein trimers. The algorithm exhaustively scans all possible combinations of contacts between three monomers, with the total number of about $10^{10}$ combinations. The running time of DockTrina is less than 2 minutes on a modern desktop computer.

To test and validate the DockTrina algorithm, we have collected two trimer docking benchmarks, consisting of 220 bound and seven unbound protein complexes, respectively. On the bound docking benchmark, our algorithm obtains 66.4% acceptable predictions listed in the top ten, 55.5% near-native predictions listed in the top ten, 52.7% acceptable predictions ranked first, and 35.9% near-native predictions ranked first. Given that Hex did not produce any acceptable pair-wise contacts for 50 cases, and any near-native contacts for 66 cases, the success rates of our algorithm are 85.9%, 79.2%, 68.2%, and 79.2%, respectively. We find that the performance of DockTrina on symmetrical proteins is similar to that of SymmDock algorithm, which was specifically developed to deal with cyclic symmetries. However, DockTrina gives significantly better results than HADDOCK and Comb Dock on protein trimers without symmetry.

We find that docking multimeric proteins is much more challenging than docking dimers. Typically, protein multimers have smaller pair-wise interface areas than dimers, making it difficult to predict individual interfaces between the monomers. On the other hand, we also find that exploiting triangular relationships between monomers provides a powerful way to identify feasible complexes. We believe it would be relatively straightforward to extend our algorithm to predict more general multimeric protein complexes with symmetry. However, given the combinatorial nature of the general assembly problem, more work will be required to model larger nonsymmetrical hetero complexes. DockTrina is available at http://nano-d.inrialpes.fr/software/docktrina or by request from the authors.

# Chapter 3

# Rapid Determination of RMSDs Corresponding to Macromolecular Rigid-body Motions

## 3.1 Introduction

The root mean square deviation is a widely used and powerful criterion to estimate the similarity between two ordered sets of points. In structural biology and bioinformatics, RMSD has been widely accepted as a measure of similarity between macromolecules. For rigid-body modeling applications, such as rigid-body molecular docking [23, 114], rigid-body molecular dynamics simulations [33, 86], and rigid-body Monte Carlo simulations [142], RMSD can be used as a measure of the rigid-body motion of a molecule. However, determination of the RMSD can be a rate-limiting step for those applications where large number of rigid-body motions should be compared. These applications range from conformation sampling in protein docking and structure-based drug design to high-throughput analysis of rigid-body modeling and simulation results.

Much effort has been spent in developing algorithms for the optimal superposition of two molecules that minimizes the RMSD between the corresponding atoms [29, 31, 35, 47, 48, 58, 61, 63, 67, 77, 81, 91, 135]. In these methods, the squared RMSD is typically minimized with respect to the components of a rotation matrix or a rotation quaternion. However, in many applications of computational chemistry and structural bioinformatics, a complementary problem emerges — given a set of rigid-body motions of a reference molecule, compute the corresponding set of RMSDs. To the best of our knowledge, there exists no explicit description of an efficient algorithm for this problem in the literature. For the case

of the RMSD between two positions of the same molecule after applying two spatial rigid-body transformations, a formula can be found in the work of Rarey et al. [112], however, it contains an error, which we correct below. Here, we present *RigidRMSD*, a new algorithm for constant time RMSD computations. In particular, we provide a connection between the RMSD and the axis and the angle of the rotation. Also, we consider rotations represented by both matrices and quaternions, since the two representations are widely used in the description of spatial transformations. We demonstrate that the quaternion representation could be more efficient than the matrix representation. Our algorithm initializes in time linear in the number of atoms in the molecule and then computes the RMSD corresponding to a rigid-body motion in constant time. The algorithm can be very useful when computing multiple RMSDs corresponding to a sequence of rigid-body motions, as, for example, in the DockTrina method [108] or clustering applications, as each new RMSD computation takes only constant time. To demonstrate the efficiency of the RigidRMSD library, we implemented an RMSD-based clustering algorithm and compared it with the standard clustering method. Finally, we provide several source-code examples that demonstrate the usage of our library.

## 3.2 Methodology

### 3.2.1 Weighted RMSD

Given two sets of $N$ points $A = \{\mathbf{a}_i\}_N$ and $A' = \{\mathbf{a}'_i\}_N$ with associated weights $\{w_i\}_N$, the weighted RMSD between them is given as

$$\mathrm{RMSD}(A,A')^2 = \frac{1}{W}\sum_i w_i \left|\mathbf{a}_i - \mathbf{a}'_i\right|^2, \tag{3.1}$$

where $W = \sum_i w_i$. Here, $\{w_i\}_N$ are statistical weights that may emphasize the importance of a certain part of the structure, for example in case of a protein, the backbone or the side chains. These weights can be also equal to atomic masses (in this case $W$ equals to the total mass of the molecule) or may be set to 1 (in this case $W = N$).

### 3.2.2 Quaternion Arithmetic

A quaternion $Q$ can be considered as a combination of a scalar $s$ with a 3-component vector $\mathbf{q} = \{q_x, q_y, q_z\}^T$, $Q = [s, \mathbf{q}]$. The product of two quaternions $Q_1 = [s_1, \mathbf{q}_1]$ and $Q_2 = [s_2, \mathbf{q}_2]$ is a quaternion and can be expressed through a combination of scalar and vector products:

$$Q_1 \cdot Q_2 \equiv [s_1, \mathbf{q}_1] \cdot [s_2, \mathbf{q}_2] = [s_1 s_2 - (\mathbf{q}_1 \cdot \mathbf{q}_2), s_1\mathbf{q}_2 + s_2\mathbf{q}_1 + (\mathbf{q}_1 \times \mathbf{q}_2)]\,. \tag{3.2}$$

The squared norm of a quaternion $Q$ is given as $|Q|^2 = s^2 + \mathbf{q}\cdot\mathbf{q}$, and a unit quaternion is a quaternion with its norm equal to 1. An inverse quaternion $Q^{-1}$ is given as $Q^{-1} = [s,-\mathbf{q}]/|Q|^2$. A vector $\mathbf{v}$ can be treated as a quaternion with zero scalar component, $\mathbf{v} \equiv [0,\mathbf{v}]$. Then, a unit quaternion $\hat{Q}$ can be used to rotate vector $\mathbf{v}$ to a new position $\mathbf{v}'$ as follows

$$\left[0,\mathbf{v}'\right] = \hat{Q}\,[0,\mathbf{v}]\,\hat{Q}^{-1} = \left[0,(s^2-\mathbf{q}^2)\mathbf{v}+2s(\mathbf{q}\times\mathbf{v})+2(\mathbf{q}\cdot\mathbf{v})\mathbf{q}\right] = [0,\mathbf{v}+2\mathbf{q}\times(\mathbf{q}\times\mathbf{v}+s\mathbf{v})]\,. \tag{3.3}$$

Equivalently, the same rotation can be represented with a rotation matrix $\mathbf{R}$, such that $\mathbf{v}' = \mathbf{R}\mathbf{v}$, where $\mathbf{R}$ can be expressed through the components of the quaternion $\hat{Q}$ as

$$\mathbf{R} = \begin{pmatrix} s^2+q_x^2-q_y^2-q_z^2 & 2q_xq_y-2sq_z & 2q_xq_z+2sq_y \\ 2q_xq_y+2sq_z & s^2-q_x^2+q_y^2-q_z^2 & 2q_yq_z-2sq_x \\ 2q_xq_z-2sq_y & 2q_yq_z+2sq_x & s^2-q_x^2-q_y^2+q_z^2 \end{pmatrix}. \tag{3.4}$$

A unit quaternion $\hat{Q}$ corresponding to a rotation by an angle $\alpha$ around a unit axis $\mathbf{u}$ is given as $\hat{Q} = [\cos\frac{\alpha}{2}, \mathbf{u}\sin\frac{\alpha}{2}]$, and its inverse is $\hat{Q}^{-1} = [\cos\frac{\alpha}{2}, -\mathbf{u}\sin\frac{\alpha}{2}]$. Finally, $N$ sequential rotations around different unit axes defined by unit quaternions $\{\hat{Q}_i\}_N$ result in a new vector $\mathbf{v}'$ according to

$$\left[0,\mathbf{v}'\right] = \hat{Q}_N\hat{Q}_{N-1}...\hat{Q}_2\hat{Q}_1\,[0,\mathbf{v}]\,\hat{Q}_1^{-1}\hat{Q}_2^{-1}...\hat{Q}_{N-1}^{-1}\hat{Q}_N^{-1}. \tag{3.5}$$

## 3.3 Rigid-body motion described with quaternions

Let $\mathbf{R}$ be a rotation matrix and $\mathbf{T}$ be a translation vector applied to a molecule with $N$ atoms at positions $A = \{\mathbf{a}_i\}_N$ with $\mathbf{a}_i = \{x_i,y_i,z_i\}^T$, such that the new positions $A' = \{\mathbf{a}'_i\}_N$ are given as $\mathbf{a}'_i = \mathbf{R}\mathbf{a}_i + \mathbf{T}$. Then, the weighted RMSD between $A$ and $A'$ is given as

$$\mathrm{RMSD}^2(A,A') = \frac{1}{W}\sum_i w_i\,|\mathbf{a}_i - \mathbf{R}\mathbf{a}_i - \mathbf{T}|^2\,. \tag{3.6}$$

We can rewrite the previous expression using quaternion representation of vectors $\mathbf{a}_i$ and $\mathbf{T}$ as

$$\mathrm{RMSD}^2 = \frac{1}{W}\sum_i w_i\left|[0,\mathbf{a}_i] - \hat{Q}[0,\mathbf{a}_i]\hat{Q}^{-1} - [0,\mathbf{T}]\right|^2. \tag{3.7}$$

Here, the unit quaternion $\hat{Q}$ corresponds to the rotation matrix $\mathbf{R}$. Since the norm of a quaternion does not change if we multiply it by a unit quaternion, we may right-multiply

the kernel of the previous expression by $\hat{Q}$ to obtain

$$\mathrm{RMSD}^2 = \frac{1}{W}\sum_i w_i \left| [0,\mathbf{a}_i]\hat{Q} - \hat{Q}[0,a_i] - [0,\mathbf{T}]\hat{Q} \right|^2. \tag{3.8}$$

Using the scalar–vector representation of a quaternion, we rewrite the previous RMSD expression as

$$\mathrm{RMSD}^2 = \frac{1}{W}\sum_i w_i \left[ -\mathbf{q}\cdot\mathbf{T}, -s\mathbf{T} + (2\mathbf{a}_i - \mathbf{T}) \times \mathbf{q} \right]^2. \tag{3.9}$$

Performing scalar and vector products in Eq. (3.9), we obtain

$$\begin{aligned} \mathrm{RMSD}^2 &= \frac{1}{W}\sum_i w_i \Big( [q_x T_x + q_y T_y + q_z T_z]^2 \\ &+ [-sT_x + q_y(2z_i - T_z) - q_z(2y_i - T_y)]^2 \\ &+ [-sT_y + q_z(2x_i - T_x) - q_x(2z_i - T_z)]^2 \\ &+ [-sT_z + q_x(2y_i - T_y) - q_y(2x_i - T_x)]^2 \Big). \end{aligned} \tag{3.10}$$

Grouping terms in Eq. (3.10) that depend on atomic positions together, we obtain

$$\begin{aligned} \mathrm{RMSD}^2 &= T_x^2 + T_y^2 + T_z^2 + \frac{4}{W}\sum_i w_i \{ q_x^2(y_i^2 + z_i^2) + q_y^2(x_i^2 + z_i^2) + q_z^2(x_i^2 + y_i^2) \\ &- 2q_x q_y x_i y_i - 2q_x q_z x_i z_i - 2q_y q_z z_i y_i \} \\ &+ \frac{4}{W}\{ q_x q_z T_z + q_x q_y T_y - q_z^2 T_x - q_y^2 T_x + s q_z T_y - s q_y T_z \} \sum_i w_i x_i \\ &+ \frac{4}{W}\{ q_y q_z T_z + q_x q_y T_x - q_x^2 T_y - q_z^2 T_y + s q_x T_z - s q_z T_x \} \sum_i w_i y_i \\ &+ \frac{4}{W}\{ q_y q_z T_y + q_x q_z T_x - q_x^2 T_z - q_y^2 T_z + s q_y T_x - s q_x T_y \} \sum_i w_i z_i. \end{aligned} \tag{3.11}$$

Introducing the inertia tensor $\mathbf{I}$, the rotation matrix $\mathbf{R}$, the center of mass (COM) vector $\mathbf{C}$, and the $3 \times 3$ identity matrix $\mathbf{E}_3$, we may simplify the previous expression to

$$\mathrm{RMSD}^2 = \mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T \mathbf{I}\mathbf{q} + 2\mathbf{T}^T (\mathbf{R} - \mathbf{E}_3)\,\mathbf{C}, \tag{3.12}$$

where $\mathbf{C} = \frac{1}{W}\{\sum w_i x_i, \sum w_i y_i, \sum w_i z_i\}^T$ , rotation matrix $\mathbf{R}$ corresponds to the rotation with the unit quaternion $\hat{Q}$ according to Eq. (3.4), and the inertia tensor $\mathbf{I}$ is given as

$$\mathbf{I} = \begin{pmatrix} \sum w_i(y_i^2+z_i^2) & -\sum w_i x_i y_i & -\sum w_i x_i z_i \\ -\sum w_i x_i y_i & \sum w_i(x_i^2+z_i^2) & -\sum w_i y_i z_i \\ -\sum w_i x_i z_i & -\sum w_i y_i z_i & \sum w_i(x_i^2+y_i^2) \end{pmatrix}. \tag{3.13}$$

Equation (3.12) is the principal result of this chapter. It consists of three parts, the pure translational contribution $\mathbf{T}^2$, the pure rotational contribution $\frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q}$, and the cross–term $2\mathbf{T}^T(\mathbf{R}-\mathbf{E}_3)\mathbf{C}$. In this equation, only two variables depend on the atomic positions $\{\mathbf{a}_i\}_N$, the inertia tensor $\mathbf{I}$, and the COM vector $\mathbf{C}$. Below, we will use this fact when computing RMSDs for a set of rigid-body motions.

### 3.3.1 RMSD Corresponding to a Pure Rotation

An interesting consequence of Eq. (3.12) is the analytical expression of the RMSD for a pure rigid-body rotation. Recall that a unit quaternion in Eq. (3.12) can be represented as a rotation about a unit axis $\mathbf{n}$ by an angle $\alpha$, $\hat{\mathbf{Q}} = [\cos\frac{\alpha}{2}, \mathbf{n}\sin\frac{\alpha}{2}]$. Then, if a rigid molecule is rotated about this axis passing through the origin, the RMSD for such a rotation is given as

$$\mathrm{RMSD}^2 = \frac{4}{W}\sin^2\frac{\alpha}{2} I(\mathbf{n}), \tag{3.14}$$

where $I(\mathbf{n})$ is the reduction of the inertia tensor (3.13) to a scalar form for the unit axis $\mathbf{n}$:

$$I(\mathbf{n}) = \mathbf{n}^T\mathbf{I}\mathbf{n}. \tag{3.15}$$

### 3.3.2 Rigid-body Motion Described with a Rotation Matrix

The pure rotational contribution $\frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q}$ in Eq. (3.12) can be rewritten in terms of a rotation matrix $\mathbf{R}$ as

$$\frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} = \frac{4}{W}\mathrm{tr}\left(\left(\mathbf{q}\mathbf{q}^T\right)\mathbf{I}\right) = \frac{1}{W}\mathrm{tr}(\mathbf{I})\left[1-\mathrm{tr}(\mathbf{R})\right] + \frac{2}{W}\mathrm{tr}(\mathbf{I}\mathbf{R}). \tag{3.16}$$

Here, rotation matrix $\mathbf{R}$ is connected with the vector part of the rotation quaternion $\mathbf{q}$ by Eq. (3.4). Equivalently, Eq. (3.16) can be written as

$$\frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} = \frac{2}{W}\sum_{i,j=1}^{3}\left(\delta_{ij}-R_{ij}\right)X_{ij}, \tag{3.17}$$

where $\delta_{ij}$ is the Kronecker delta and matrix $\mathbf{X}$ is given as

$$\mathbf{X} = \begin{pmatrix} \sum w_i x_i^2 & \sum w_i x_i y_i & \sum w_i x_i z_i \\ \sum w_i x_i y_i & \sum w_i y_i^2 & \sum w_i y_i z_i \\ \sum w_i x_i z_i & \sum w_i y_i z_i & \sum w_i z_i^2 \end{pmatrix}. \tag{3.18}$$

Now, the weighted RMSD in Eq. (3.12) can be computed using the matrix description of the rotation:

$$\mathrm{RMSD}^2 = \mathbf{T}^2 + \frac{2}{W} \sum_{i,j=1}^{3} \left(\delta_{ij} - R_{ij}\right) X_{ij} + 2\mathbf{T}^T \left(\mathbf{R} - \mathbf{E}_3\right) \mathbf{C}. \tag{3.19}$$

### 3.3.3 RMSD Corresponding to a Relative Rigid-body Motion

Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be two rotation matrices and $\mathbf{T}_1$ and $\mathbf{T}_2$ – two translation vectors applied to a molecule with $N$ atoms at positions $A = \{\mathbf{a}_i\}_N$, such that new positions $A_1 = \{\mathbf{a}_i^1\}_N$ and $A_2 = \{\mathbf{a}_i^2\}_N$ are given as $\mathbf{a}_i^1 = \mathbf{R}_1\mathbf{a}_i + \mathbf{T}_1$ and $\mathbf{a}_i^2 = \mathbf{R}_2\mathbf{a}_i + \mathbf{T}_2$. Let a unit quaternion $\hat{Q} = [s, \mathbf{q}]$ correspond to the relative rotation $\mathbf{R}_2^T\mathbf{R}_1$. Then, the weighted RMSD between positions $A_1$ and $A_2$ is given by a generalized version of Eq. (3.12) as

$$\mathrm{RMSD}^2(A_1, A_2) = \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} + (\mathbf{T}_1 - \mathbf{T}_2)^2 + 2\left(\mathbf{T}_1 - \mathbf{T}_2\right)^T \left(\mathbf{R}_1 - \mathbf{R}_2\right) \mathbf{C}. \tag{3.20}$$

Using Eq. (3.17) we can rewrite the above equation using the matrix description of the rotation:

$$\mathrm{RMSD}^2(A_1, A_2) = \frac{2}{W} \sum_{i,j=1}^{3} \left( \delta_{ij} - \sum_{k=1}^{3} R_{ki}^1 R_{kj}^2 \right) X_{ij} + (\mathbf{T}_1 - \mathbf{T}_2)^2 + 2\left(\mathbf{T}_1 - \mathbf{T}_2\right)^T \left(\mathbf{R}_1 - \mathbf{R}_2\right) \mathbf{C}. \tag{3.21}$$

The derived equation is equivalent to the formula obtained by Rarey et al. for clustering spatial motions in the FlexX docking tool [112], except that the formula of Rarey et al. contains an error in the rotational part. More precisely, it has an additional factor 2 preceding the $\sum_{k=1}^{3} R_{ki}^1 R_{kj}^2$ term.

## 3.4 Algorithm Implementation

### 3.4.1 Computational Considerations

In the above Eqs. (3.12) — (3.21) , as we have mentioned earlier, only two variables depend on the atomic positions of the reference molecular structure — the inertia tensor $\mathbf{I}$ (or, its

equivalent matrix $X$ if the rotation is given using the matrix representation) , and the COM vector $\mathbf{C}$. Therefore, given a set of $M$ spatial transformations, we compute these two variables only once at the initialisation step. The computational complexity of this step is linear with respect to the number of atoms $N$ in the molecule. After, each RMSD computation for a single spatial transformation takes only constant time. The total cost to compute $M$ RMSD values for a rigid molecule with $N$ atoms thus will be $O(N+M)$, which is usually much smaller compared to the cost of standard algorithms, $O(NM)$, particularly at large values of $M$ and $N$. More precisely, a standard algorithm computes the RMSD for each spatial transformation in $O(N)$ operations according to Eq. (3.1), thus resulting in $O(NM)$ overall complexity for M spatial transformations. Below we discuss computational strategies that allow to reduce the constant in $O(N+M)$.

In Eq. (3.12), the cross–term vanishes in the reference frame bound to the COM of the molecule where $\mathbf{C} = \mathbf{0}$. In this reference frame, the rotation is preserved, while the translation $\mathbf{T}_{\text{COM}}$ is given as

$$\mathbf{T}_{\text{COM}} = \mathbf{RC} + \mathbf{T} - \mathbf{C}. \tag{3.22}$$

We can equivalently obtain the translation in the COM reference frame using a rotation quaternion $\hat{Q}$ as

$$\mathbf{T}_{\text{COM}} = \hat{Q}\mathbf{C}\hat{Q}^{-1} + \mathbf{T} - \mathbf{C}. \tag{3.23}$$

Therefore, in the COM reference frame, the RMSD can be computed with fewer arithmetic operations. More precisely, using quaternion representation of the rotation, the RMSD is given as

$$\text{RMSD}^2 = \mathbf{T}^2_{\text{COM}} + \frac{4}{W}\mathbf{q}^T\mathbf{I}_{\text{COM}}\mathbf{q}. \tag{3.24}$$

Similarly, if we use matrix representation of the rotation, the RMSD is given as

$$\text{RMSD}^2 = \mathbf{T}^2_{\text{COM}} + \frac{2}{W}\sum_{i,j=1}^{3}\left(\delta_{ij} - R_{ij}\right)X^{\text{COM}}_{ij}. \tag{3.25}$$

In the above equations, inertia tensor $\mathbf{I}_{\text{COM}}$ and matrix $\mathbf{X}^{\text{COM}}$ are computed in the COM coordinate system. A particularly interesting case is the computation of the RMSD in the principal axes of inertia (PAI) frame. The PAI frame is the coordinate system where the centre of mass vector $\mathbf{C} = \mathbf{0}$ and the molecule is aligned along its principal axes, that is, matrices $\mathbf{I}_{\text{COM}}$ and $\mathbf{X}^{\text{COM}}$ are diagonal. In this frame, Eqs. (3.24) and (3.25) are simpler. Also, in the PAI frame, RMSD corresponding to a relative rigid-body motion defined by

two rotation quaternions $\hat{Q}_1$ and $\hat{Q}_2$ and two translation vectors $\mathbf{T}_1$ and $\mathbf{T}_2$ will be

$$\begin{aligned}\mathrm{RMSD}^2(A_1,A_2) = \frac{4}{W}\big(&\left(s_1 q_2^x - q_1^x s_2 - q_1^y q_2^z + q_1^z q_2^y\right)^2 I_{xx} + \left(s_1 q_2^y - q_1^y s_2 - q_1^z q_2^x + q_1^x q_2^z\right)^2 I_{yy} + \\ &\left(s_1 q_2^z - q_1^z s_2 - q_1^x q_2^y + q_1^y q_2^x\right)^2 I_{zz}\big) + (\mathbf{T}_1 - \mathbf{T}_2)^2 . \end{aligned} \tag{3.26}$$

This equation uses three times fewer arithmetic operations compared to the previously published Eq. (3.21). More precisely, Eq. (3.26) requires only 38 arithmetic operations compared to 114 operations in Eq. (3.21).

Generally, Eqs. (3.22) — (3.26) are more efficient in the number of arithmetic operations compared to Eqs. (3.12) and (3.21), as it is summarized in Table 3.1. This table lists the

Table 3.1 Number of arithmetic operations for the squared RMSD calculations with respect to different rotation representations and a different choice of the coordinate frame. These numbers were computed according to the source code of the RigidRMSD library. The references to the corresponding equations are given in the last column. These equations comprise only multiplication and addition/subtraction arithmetic operations.

| | multiplies | add/subtract | Total | Equation |
|---|---|---|---|---|
| $\mathrm{RMSD}^2$ (quaternion, world frame) | 34 | 20 | 54 | (3.24) and (3.23) |
| $\mathrm{RMSD}^2$ (matrix, world frame) | 19 | 26 | 45 | (3.25) and (3.22) |
| $\mathrm{RMSD}^2$ (quaternion, COM frame) | 16 | 8 | 24 | (3.24) |
| $\mathrm{RMSD}^2$ (matrix, COM frame) | 10 | 14 | 24 | (3.25) |
| $\mathrm{RMSD}^2$ (quaternion, PAI frame) | 9 | 5 | 14 | (3.24), $\mathbf{I}_{\mathrm{COM}}$ is diagonal |
| $\mathrm{RMSD}^2$ (matrix, PAI frame) | 6 | 8 | 14 | (3.25), $\mathbf{X}^{\mathrm{COM}}$ is diagonal |
| $\mathrm{RMSD}^2$ for clustering, (matrix, world frame) | 55 | 59 | 114 | (3.21) |
| $\mathrm{RMSD}^2$ for clustering, (quaternion, PAI frame) | 21 | 17 | 38 | (3.26) |

number of arithmetic operations needed to compute the squared RMSD using different representations of the rigid-body motion in three different coordinate systems, the world frame, the COM frame, and the PAI frame. As listed in Table 3.1, to compute the squared RMSD we need 54 arithmetic operations in the worst case, when the rigid-body rotation is given as a quaternion in the world frame. If we choose the coordinate system properly (the PAI

frame), we can compute the squared RMSD in just 14 operations. Table 3.1 demonstrates that in the world frame one requires a fewer number of arithmetic operations to compute the RMSD if rotations are represented with rotation matrices, whereas in the COM and PAI frames the number of operations is equal between the two representations. However, when performing sequences of rotations, the quaternion representation is more numerically stable and computationally efficient compared to the matrix representation irrespective of the choice of coordinate system. Indeed, one requires 45 arithmetic operations to multiply two rotation matrices, whereas quaternion multiplication requires only 28 operations. Finally, Table 3.1 demonstrates that the squared RMSD for a relative rigid-body motion computed with the quaternion representation in the PAI frame requires three times fewer operations compared to the one computed with the matrix representation in the world frame.

### 3.4.2 Numerical Tests

Throughout the article, we count the number of arithmetic operations in different equations according to the source code of the RigidRMSD library. We would like to mention that the cost of different arithmetic operations is not the same - division and square root are usually more expensive than multiplication, which is in turn more expensive than addition and subtraction [18]. We should also mention that on modern computers minimizing the number of arithmetic operations is less important for the performance of a particular algorithm compared to increasing the amount of instruction level parallelism or improving memory access patterns and cache utilization, for example. Therefore, it is impossible to rigorously compare the performance of different algorithms solely based on their operation count. Thus, we only provide the total number of arithmetic operations as a rough estimation of the complexity of the equations and the corresponding algorithms. To get more practical numbers, in the following sections we run a series of tests with two different levels of compiler optimization.

We implemented the tests using the C++ programming language and compiled them using g++ compiler version 4.6 with optimization levels -O0 and -O3. For the gcc family of compilers, optimization option -O0 disables compiler optimization, whereas optimization option -O3 enables heavy optimization including inter-procedural optimization and vectorization. We ran the tests on a 64-bit Linux Fedora operating system with Intel(R) Xeon(R) CPU X5650 @ 2.67GHz.

## 3.5 Results and Discussion

This section presents numerical tests and practical applications of the equations derived in this article. First, we compare the quaternion representation with the matrix representation when computing sequential rotations (i.e., a composition of several rotations) and when computing a product of rotations with the subsequent RMSD computation. Second, we discuss the similarity measure between molecules and demonstrate that the rotation RMSD (see Eq. (3.14)) can be advantageous over a simpler angular distance measure. Finally, we present a rigid-body clustering algorithm as an example of the application of the derived equations.

### 3.5.1 Rotation Representation

Quaternions provide another way to represent rotations compared to conventional rotation matrices. In practice, the quaternion representation has several benefits over the matrix representation. First, a quaternion compared to a matrix requires less storage, four values versus nine. Second, the orthonormalization of a quaternion costs much less than the orthogonalization of a matrix. More precisely, orthonormalization of a quaternion can be accomplished by dividing the quaternion by its norm, which requires twelve arithmetic operations including one square root. However, there is no universal method for matrix orthonormalization. In this case, one may use the Gram–Schmidt orthonormalization method, QR decomposition, singular value decomposition or other methods, which are more computationally expensive compared to the quaternion orthonormalization [39]. Third, a product of two rotations using quaternions requires fewer arithmetic operations compared to the matrix representation (28 versus 45). Finally, the matrix multiplication is less numerically stable due to the accumulation of rounding errors. In summary, applications that require sequential rotations (e.g., some docking applications) will gain in speed, memory, and numerical precision when using the quaternion representation.

To demonstrate the numerical efficiency of the quaternion representation, we ran a series of tests with two different levels of compiler optimization. In the first test, we performed $10^8$ products of rotations using the two types of rotation representation and compared the timing for a single product of rotations with and without compiler optimization. The results of this test are presented in Table 3.2. We see that a rotation with quaternions is about 60with matrices regardless of the optimization level. In the second test, we computed a product of two rotations with the subsequent RMSD computation using Eqs. (3.22) — (3.25) and repeated these operations $10^8$ times. Then, we calculated the time required for a single product of rotations with the subsequent RMSD computation. The results of this test are

also presented in Table 3.2. Again, the quaternion representation is about 10% faster without optimization and 4% faster with optimization compared to the matrix representation. We should note that increasing the number of sequential rotations will provide a bigger speedup using the quaternion representation in this example. In the third test, we computed $10^8$ RMSDs corresponding to a relative rigid-body motion using the matrix representation of rotation (see Eq. (3.21)) and the quaternion representation of rotation (see Eq. (3.26)). We can see that our quaternion approach is 2.4–3.2 times faster compared to the matrix formula (see Eq. (3.21)) depending on the level of the hardware optimization.

Table 3.2 Running time for three tests using two levels of compiler optimization. O0 optimization level disables optimization, whereas O3 optimization level enables heavy optimization including interprocedural optimization and vectorization. In the first test (columns 1 and 2), we performed $10^8$ products of rotations using the two types of rotation representation and reported the timing for a single product of rotations. In the second test (columns 3 and 4), we computed a product of two rotations with the subsequent RMSD computation using Eqs. (3.22) — (3.25) and repeated these operations $10^8$ times for averaging. In the last test (columns 5 and 6), we computed $10^8$ RMSDs corresponding to a relative rigid-body motion, as in the clustering application, using the matrix representation of rotation (see Eq. (3.21)) and the quaternion representation of rotation (see Eq. (3.26)) and reported the timing for a single RMSD calculation.

| | Product of Rotations (-O0) | Product of Rotations (-O3) | Rotations and RMSD (-O0) | Rotations and RMSD (-O3) | Clustering (-O0) | Clustering (-O3) |
|---|---|---|---|---|---|---|
| Quaternion representation | $2.96\times 10^{-8}$ s | $0.73\times 10^{-8}$ s | $7.79\times 10^{-8}$ s | $2.29\times 10^{-8}$ s | $4.17\times 10^{-8}$ s | $1.19\times 10^{-8}$ s |
| Matrix representation | $4.68\times 10^{-8}$ s | $1.18\times 10^{-8}$ s | $8.55\times 10^{-8}$ s | $2.39\times 10^{-8}$ s | $9.99\times 10^{-8}$ s | $3.81\times 10^{-8}$ s |

To summarize, if a particular application operates with sequential rotations, as it happens in the DockTrina algorithm [108] or other docking applications, RMSD computations are more numerically efficient using the quaternion representation. Furthermore, the gain of using the quaternion representation is bigger up to 60% when using a larger sequence of rotations.

### 3.5.2 Rotation RMSD as a similarity measure for molecular structures

It is still an open question how to measure the similarity between structures of a molecular complex [147]. For example, Rodrigues et al[117] developed a clustering method with the similarity measure based on the fraction of common contacts between two complexes.

Another similarity measure was recently proposed by Vreven et al. [146], where the angular distance computed in constant time is used as the criterion for the similarity between the predictions from rigid-body docking. Nonetheless, the majority of the algorithms in the structural bioinformatics use the pair-wise RMSD as the similarity metric between the molecular structures.

Equation (3.14) is of particular interest when considered in relation to the aforementioned work of Vreven et al. [146], where the authors demonstrated that the angular distance can serve as a similarity measure for rigid molecules as an alternative for the RMSD. More precisely, they defined the angular distance as the angle between the rotations corresponding to two docking predictions, ignoring the translational degrees of freedom. Vreven et al. claimed that the drawback of using the RMSD is that it is computationally expensive. However, we demonstrated that the RMSD can also be computed in constant time. Furthermore, in the context of Eq. (3.14), the angular distance is simply equal to the rotation angle $\alpha$ In particular, for a fixed rotation angle, the angular distance for molecules of different size will be equal, while the RMSD can be very different. Another example that demonstrates the difference between the two measures is the rotation of a long linear molecule. The RMSD for such a rotation will dramatically depend on the axis of the rotation, while the angular distance will be the same regardless the rotation axis.

To conclude, we would like to emphasize that for comparison of rigid molecules of different size or molecules of non-spherical shape, it may be more rigorous to use the similarity measure defined by Eq. (3.14) instead of the angular distance. Particularly, our measure involves the scalar form of the inertia tensor (see Eq. (3.15)), thus taking into account the geometry and the rotation axis of the molecules.

### 3.5.3 Clustering

One of the possible applications of the RigidRMSD library can be the rigid-body clustering. Molecular docking algorithms typically produce thousands of solutions, some of them having a very similar geometry. Therefore, it is practical to group these into clusters. As we have discussed above, there are multiple ways to measure the similarity between molecular structures [147], however, most of the modern state-of-the-art clustering algorithms use the pair-wise RMSD as the similarity metric between the predictions, as it is implemented, for example, in the Hex [114] and ZDOCK [23] docking algorithms. In the worst case, the complexity of such a clustering algorithm can be quadratic with respect to the number of docking predictions. Thus, an efficient pair-wise RMSD test can dramatically improve the clustering performance. The clustering algorithm used by the Hex and ZDOCK applications consists of the following steps. First, the docking prediction with the best score (yet unas-

signed to any cluster) is taken as the seed for the new cluster. Second, the pair-wise RMSDs between the seed and all other predictions (in case of ZDOCK) or some best predictions (in case of Hex) are measured and the predictions with the RMSD lower than a certain threshold are put into the cluster. Finally, these two steps are iterated until all docking predictions are assigned to corresponding clusters.

To demonstrate the efficiency of the RigidRMSD library, we compared the clustering algorithm implemented with our library to the one from the Hex software. We chose Hex for the comparison because it is a very fast rigid-body docking tool and also because it explicitly provides the clustering time. It is worth to note that Hex's clustering algorithm has linear complexity with respect to the number of docking predictions, that is, it is faster (although less accurate) than the standard RMSD-based clustering algorithms, as it is implemented in ZDOCK. Both Hex and ZDOCK clustering algorithms use the standard RMSD test linear in the number of atoms in the protein.

For the comparison, we collected a benchmark of 23 protein dimers of various size (see Table 3.3). After, we launched Hex version 6.3 on this benchmark and collected docking solutions before clustering, sizes of clusters, and clustering time. We then also clustered these solutions using the RigidRMSD library. Figure 3.1 shows the clustering time of the Hex clustering algorithm with respect to our clustering using Eqs. (3.21) and (3.26) as a function of the number of atoms in the smaller protein (left) and the number of docking solutions before the clustering (right). We can clearly see that our implementation of the clustering algorithm is more than an order of magnitude faster compared to the Hex implementation. Also, the quaternion representation of rotation, Eq. (3.26), is on average three times more efficient compared to the matrix representation, Eq. (3.21). The efficiency of our clustering algorithm increases when using a larger RMSD threshold, as it is shown in Fig. 3.2. Also, mean cluster sizes obtained with our clustering algorithm are significantly larger compared to the Hex clustering (see Fig. 3.2), particularly at large RMSD thresholds. This demonstrates that our implementation of the clustering algorithm is not only much faster, but also more accurate compared to the clustering in Hex, especially at large clustering thresholds.

## 3.6 Conclusions

We described a very fast and efficient way to compute the RMSD corresponding to the set of rigid-body motions of a molecule. Our algorithm consists of an initialization step followed by a series of constant time RMSD computations. The initialization step has linear complexity with respect to the number of atoms in a molecule. However, each of the RMSD calculations requires only 14 to 54 arithmetic operations when using a single rigid-body

Table 3.3 Benchmark of protein dimers. First two columns represent names of protein monomer in a protein complex according to PDB. The third column lists the number of atoms in the smaller protein.

| First protein | Second protein | Number of atoms in the smaller protein |
|---|---|---|
| 1AYM::A | 1AYM::B | 1981 |
| 1AYM::B | 1AYM::C | 1847 |
| 1AYM::C | 1AYM::A | 2336 |
| 1B35::A | 1B35::B | 1412 |
| 1B35::B | 1B35::C | 2129 |
| 1B35::C | 1B35::A | 2029 |
| 1EPT::A | 1EPT::B | 768 |
| 1EPT::B | 1EPT::C | 863 |
| 1EPT::C | 1EPT::A | 388 |
| 1RM6::A | 1RM6::B | 2422 |
| 1RM6::B | 1RM6::C | 1178 |
| 1SR4::A | 1SR4::B | 2025 |
| 1SR4::B | 1SR4::C | 1203 |
| 1SR4::C | 1SR4::A | 1308 |
| 1W85::A | 1W85::B | 2569 |
| 1W85::B | 1W85::C | 2483 |
| 1W85::C | 1W85::A | 2473 |
| 2WJN::A | 2WJN::B | 2161 |
| 2WJN::B | 2WJN::C | 2451 |
| 2WJN::C | 2WJN::A | 1876 |
| 3VBH::A | 3VBH::B | 2300 |
| 3VBH::B | 3VBH::C | 1896 |
| 3VBH::C | 3VBH::A | 1863 |

motion (i.e., given with a single spatial rigid-body transformation), or 38 to 114 arithmetic operations when using a relative rigid-body motion (i.e., given with a pair of spatial rigid-body transformations), depending on the representation of the motion and the choice of the coordinate frame. This can be compared to 30 arithmetic operations needed to rotate a vector using a quaternion or 15 arithmetic operations needed to rotate a vector using a rotation matrix. We demonstrated that RMSD computations are more numerically efficient when using the quaternion representation of rotation. In particular, the gain of using the quaternion representation is bigger when using a larger sequence of rotations. We have also discussed two ways to measure the similarity between structures of a molecular complex. Specifically, we claim that it may be more rigorous to use the rotation RMSD similarity measure defined by (3.14) instead of the simpler measure based on the angular distance. As an application of

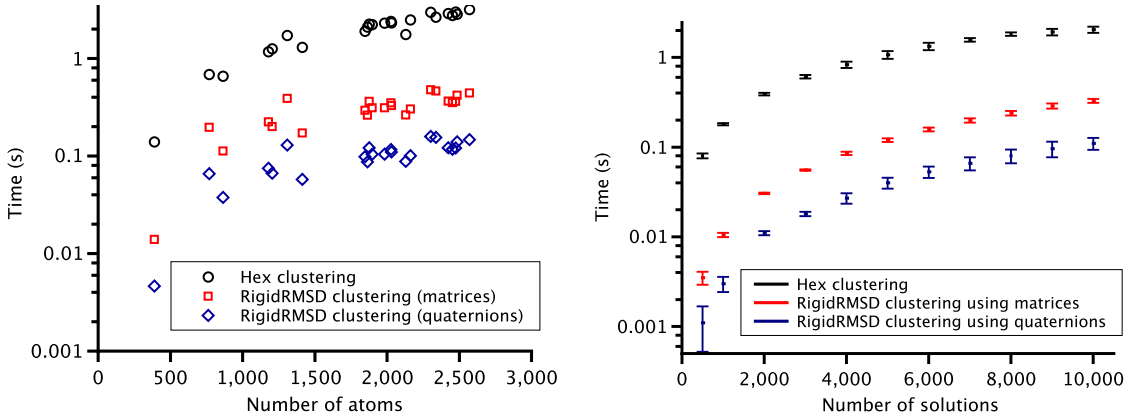


Figure 3.1 Left: Time spent on clustering docking solutions by Hex and RigidRMSD with respect to the number of atoms in the smaller protein. Each point on the plot corresponds to a protein complex from the protein benchmark (see Table 3.3). For each protein complex, the number of considered docking solutions was fixed to 10,000. Right: Average time spent on clustering docking solutions by Hex and RigidRMSD with respect to the number of docking solutions. For this plot, we chose five structures with the number of atoms in the smaller protein of about 2000 such that they result in a similar number of clusters and plotted the standard deviation of the clustering time for these structures. For both plots, time is plotted on a logarithmic scale and the clustering RMSD threshold is fixed to 10.0 Å.

the RigidRMSD library, we implemented a clustering algorithm for solutions obtained with rigid-body molecular docking tools. We showed that our implementation is more than one order of magnitude faster and also more accurate compared to the standard clustering algorithm used in the popular Hex docking software. A C++ implementation of the RigidRMSD library is available at http://nano-d.inrialpes.fr/software/RigidRMSD.

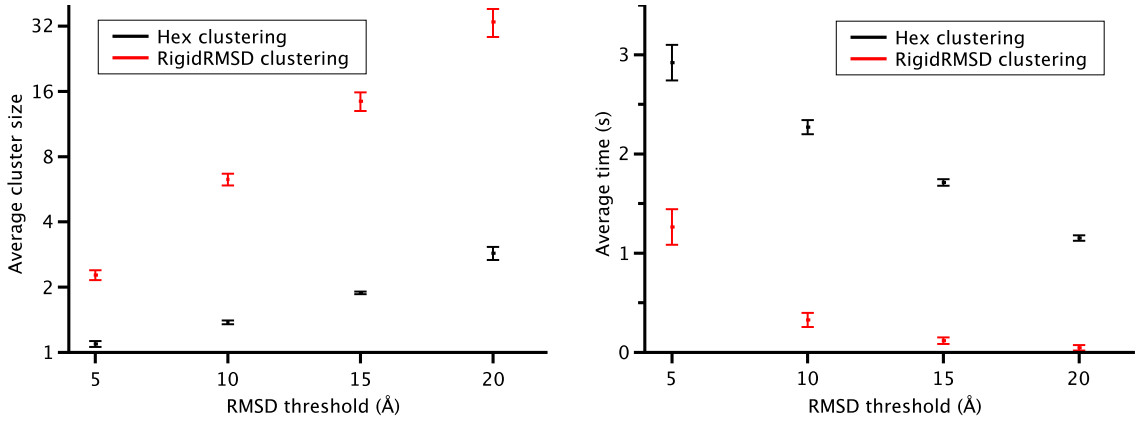


Figure 3.2 Left: Average size of a cluster provided by Hex and RigidRMSD with respect to the RMSD cluster threshold. Right: Average time spent on clustering docking solutions by Hex and RigidRMSD with respect to the RMSD cluster threshold. For both plots, we chose five structures with the number of atoms in the smaller protein of about 2000 such that they result in a similar number of clusters. For each protein complex, the number of considered docking solutions was fixed to 10,000.

# Chapter 4

# Knowledge-based Scoring Function for Protein-Protein Interactions

## 4.1 Introduction

Protein-protein interactions play crucial role in the human interactome, orchestrating most of the signaling network processes. Abrupt changes in protein-protein interactions lead to various kind of diseases, which makes protein structure prediction an important challenge in rational drug design. However, generally it is very difficult to experimentally obtain structures of protein complexes, thus computational molecular docking techniques are often used nowadays for protein-protein structure prediction. Typically, molecular docking as an integral part of the drug discovery process involves the scoring stage, where one selects the best putative binding candidates from the set of binding poses by assigning the score or the energy value $E$ to each candidate. The scoring stage incorporates sophisticated scoring functions [95], which are obtained with the empirical force-fields or using information derived from experimentally obtained structures of protein complexes. The latter type of scoring functions belongs to the family of the *knowledge-based* or *statistical* scoring functions. The majority of modern knowledge-based scoring functions for the protein-protein interactions are developed following the observation that the distances between the atoms in experimentally determined structures follow the Boltzmann distribution [36]. More precisely, using ideas from statistical theory of liquids, effective potentials between atoms are extracted using the inverse Boltzmann relation, $E_{ij}(r) = -k_B T \log(P_{ij}(r)/Z)$, where $k_B$ is the Boltzmann constant, $P_{ij}(r)$ denotes the probability to find two atoms of certain types $i$ and $j$ at a distance $r$, and $Z$ denotes the probability distribution in the reference state. The latter is the thermodynamic equilibrium state of the protein when all interactions between

the atoms are set to zero. The score of a protein conformation is then given as a sum of the effective potentials between all pairs of atoms. Although this concept is old and originates from the work of Tanaka and Scheraga [134], Miyazawa and Jernigan [93] and Sippl [130], it is still under debates [9, 70, 131, 136]. Particularly, the computation of the reference state is a challenging problem [76]. Although some assumptions were made to ease the expression of the reference state for protein monomers [82, 121, 130, 153], to deduce scoring functions for the protein-protein docking, one usually computes the reference state based on a large set of generated non-native conformations of protein complexes (decoys). [25, 51]. Another type of statistical potentials is constructed using the discriminative machine learning, specifically, the linear programming approach [3, 22, 84, 110, 111, 113, 138]. The basic idea behind this approach is to solve a system of inequalities that demand the energy of the native conformation to be lower than the energy of all the decoy conformations for a particular complex, $E(P^{\text{native}}) - E(P_i^{\text{decoy}}) < 0,\ \forall P_i^{\text{decoy}} \in \mathbf{P}^{\text{decoy}}$. Although this approach circumvents the computation of the reference state, its success critically depends on the chosen set of decoy conformations $\mathbf{P}^{\text{decoy}}$. Thereby, the obtained statistical potential depends on the sampling algorithm used to generate the decoy conformations and, generally, might not distinguish the native structures equally well from decoys obtained by another sampling algorithm.

In this study we discovered that knowledge of only native protein-protein interfaces is sufficient to construct well-discriminative predictive models for the selection of putative binding candidates. Namely, we introduce a new scoring method that comprises a knowledge-based scoring function called KSENIA deduced from the structural information about the native interfaces of 844 crystallographic protein-protein complexes. As a result, our approach does not require neither the computation of the reference state nor the ensemble of non-native complexes. Thus, it has no bias toward a method to generate putative binding poses. To the best of our knowledge, this is the first investigation of the knowledge-based scoring function that needs no information derived from non-native protein-protein interfaces. More precisely, we use convex optimization to train the knowledge-based scoring function on sets of near-native conformations with the average root mean square deviation (RMSD) between monomers of 1 Å. These are composed using the deformations along the directions of low-frequency normal modes computed at the native conformations. We demonstrate that the obtained scoring function is capable to distinguish the native and near-native protein-protein interactions from the non-native ones. Given that rigid-body minimization refinement improves the scoring performance [92], we also implement a rigid-body optimization protocol using the derived knowledge-based potential. Finally, we verify the robustness of our method on several protein-protein docking benchmarks.

## 4.2 Theoretical Basis

We consider $N$ native protein-protein complex conformations $P_i^{\text{native}}$, $i = 1..N$. For each protein complex $i$ we generate $D$ decoys, $P_{ij}^{\text{decoy}}$, $j = 1..D$, where the first index runs over different protein complexes and the second index runs over generated decoys. Then we find a linear *scoring functional* $F$, defined for all possible complexes, such that for each native complex $i$ and its decoy $j$ the following inequality holds:

$$F(P_i^{\text{native}}) < F(P_{ij}^{\text{decoy}}) \tag{4.1}$$

We express the scoring functional which fulfills these assumptions in the following form:

$$F(P) = \sum_{k=1}^{M} \sum_{l=k}^{M} \int_0^{r_{\max}} n^{kl}(r) U^{kl}(r)\, dr, \tag{4.2}$$

where $n^{kl}(r)$ is the *number density* of atom pairs at a distance $r$ between two atoms of types $k$ and $l$ ($kl$-pair), with one atom located in the larger protein (receptor), and the other atom located in the smaller protein (ligand). Here, $M$ is the total number of different atom types. We used $M = 20$ atom types definitions provided by Huang and Zou [51], which were defined by the classification of all heavy atoms in standard amino acids according to their element symbol, aromaticity, hybridization, and polarity. The functions $U^{kl}(r)$ are unknown *scoring potentials*, which we determine below.The number density $n^{kl}(r)$ is computed as a sum over all $kl$-pairs in a given protein complex via:

$$n^{kl}(r) = \sum_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-r_{ij})^2}{2\sigma^2}} \tag{4.3}$$

Here, each $kl$-pair at a distance $r_{ij}$ is represented by a Gaussian centered at $r_{ij}$ with the standard deviation of $\sigma$, which takes into account possible inaccuracies and thermal fluctuations in the protein structure. In our work we chose $\sigma = 0.4$ Å, since this value demonstrated the best results in the cross-validation tests (see Section 4.3.2 for more details). We considered only atom pairs at distances below the threshold distance $r_{\max} = 10$ Å. Using Eq. (4.3), we can re-write the scoring functional $F(P)$ (see Eq. (4.2)) as the sum over all $kl$-pairs of atoms $i$ and $j$ at a distance $r_{ij}$:

$$F(P) = \sum_{ij} \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^{r_{\max}} e^{-\frac{(r-r_{ij})^2}{2\sigma^2}} U^{kl}(r)\, dr = \sum_{ij} \Upsilon^{kl}(r_{ij}) \tag{4.4}$$

We will refer to the functions

$$\Upsilon^{kl}(r) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^{r_{\max}} e^{-\frac{(x-r)^2}{2\sigma^2}} U^{kl}(x)\, dx, \tag{4.5}$$

which are the Gauss transform of the scoring potentials $U^{kl}(x)$, as to the *scoring functions*.

In order to determine unknown scoring potentials $U^{kl}(r)$ (see Eq. (4.2)), we decompose them along with the number densities $n^{kl}(r)$ in a polynomial basis:

$$\begin{aligned} U^{kl}(r) &= \sum_q w_q^{kl} \psi_q(r), \quad r \in [0; r_{\max}] \\ n^{kl}(r) &= \sum_q x_q^{kl} \psi_q(r), \quad r \in [0; r_{\max}], \end{aligned} \tag{4.6}$$

where $\psi_q(r)$ are orthogonal basis functions on the interval $[r_1; r_2]$, and $w_q^{kl}$ with $x_q^{kl}$ are the expansion coefficients of $U^{kl}(r)$ and $n^{kl}(r)$, respectively. Here, we use a set of shifted rectangular functions as the basis [30]. Given this, the scoring functional $F$ (see Eq. (4.2)) can be expanded up to the order $Q$ as:

$$F(P) \approx \sum_{k=1}^{M} \sum_{l=k}^{M} \sum_{q}^{Q} w_q^{kl} x_q^{kl} = (\mathbf{w} \cdot \mathbf{x}), \quad \mathbf{w}, \mathbf{x} \in \mathbb{R}^{Q \times M \times (M+1)/2} \tag{4.7}$$

We will refer to the vector $\mathbf{w}$ as to the *scoring vector* and to the vector $\mathbf{x}$ as to the *structure vector*. Then, we can re-write the set of inequalities ((4.1)) as a *soft-margin quadratic optimization problem* [17]:

$$\begin{aligned} &\text{Minimize (in } \mathbf{w}, b_i, \xi_{ij}\text{):} && \tfrac{1}{2}\mathbf{w}\cdot\mathbf{w} + \textstyle\sum_{ij} C_{ij}\xi_{ij} \\ &\text{Subject to:} \\ & && y_{ij}\left[\mathbf{w}\cdot\mathbf{x}_{ij} - b_i\right] - 1 + \xi_{ij} \geq 0,\ i = 1..N,\ j = 0..D \\ & && \xi_{ij} \geq 0 \end{aligned} \tag{4.8}$$

Here, index $i$ runs over different protein complexes and index $j$ runs over different conformations of the $i$-th protein complex. Particularly, protein conformations with $j = 0$ are native with the corresponding constants $y_{i0} = +1$ and protein conformations with $j = 1..D$ are the decoys with the corresponding constants $y_{ij} = -1$. Parameters $C_{ij}$ can be regarded as regularization parameters, which control the importance of different structure vectors. We found the optimal values of $C_{ij}$ parameters using the cross-validation procedure (see Section 4.3.2). The scoring vector $\mathbf{w}$, the offset vector $\mathbf{b}$ and the slack variables $\xi_{ij}$ are the parameters to be optimized. The size of the optimization problem is determined by the dimensionality

of the structure and the scoring vectors, which is equal to $Q \times M \times (M+1)/2 = 8400$, and by the size of the training set, $N = 844$ and $D = 225$. The latter is composed based only on local information about the native interfaces of protein-protein complexes and no other information is used (see Section 4.3.5). We solve problem ((4.8)) in its *dual form* using the *block sequential minimal optimization* (BSMO) algorithm as explained elsewhere [30]. Finally, given the solution of problem (4.8), i.e. the scoring vector $\mathbf{w}$, one may restore the scoring potentials $U^{kl}(r)$ (see Eq. (4.6) ), the scoring functions $\Upsilon^{kl}(r)$ (see Eq. (4.5)), and compute the score of a protein complex according to Eq. (4.4).

## 4.3 Material and Methods

### 4.3.1 Artificial Potential Barriers

To reconstruct potential barriers at short distances, we first introduce two *barrier support* points, $P_1$ and $P_2$. The first point $P_1(0,100)$ defines the height of the potential barrier. The second point $P_2(x_{\text{rdf}},77)$ defines the width of the barrier. The distance $x_{\text{rdf}}$ varies for different $kl$-pairs of atoms and is determined as the abscissa of the first point with a non-zero $y$-coordinate of the radial distribution function corresponding to the $kl$-pair, which is computed from the native structures in the training set. Then, we classify each scoring function $\Upsilon^{kl}(r)$ (see Eq. 4.5) as *steep* or *flat*:

$$\Upsilon^{kl}(r)\text{is}: \begin{cases} \textit{steep} & \text{, if } x_{\max} \leq 5\text{Å and } \Delta y \geq 1.5 \\ \textit{flat} & \text{, otherwise} \end{cases},$$

where $x_{\max}$ is the $x$-coordinate of the first local maximum of a potential $\Upsilon^{kl}(r)$ and $\Delta y$ is the difference between $y$-coordinates of the first local maximum and minimum.

After, we discretize the functions $\Upsilon^{kl}(r)$ with 40 *support* points, positioned evenly in x. Later, we use a cubic spline interpolation through these points to reconstruct the original functions. To introduce the artificial barriers, we replace some of the support points prior to the interpolation. More precisely, for the *steep* scoring functions we remove all the support points in the interval of $[0,x_b]$, with $x_b$ being the *inflection* point between the first local maximum and minimum, where the second derivative of the scoring function changes its sign from negative to positive. For the *flat* scoring functions, we remove all the support points in the interval of $[0,x_b]$, where $x_b = 1.35x_{\text{rdf}}$. Finally, we replace all the support points in the defined intervals by the two *barrier support points* $P_1$ and $P_2$.

At the very last step we use the cubic spline interpolation to smoothly delineate the

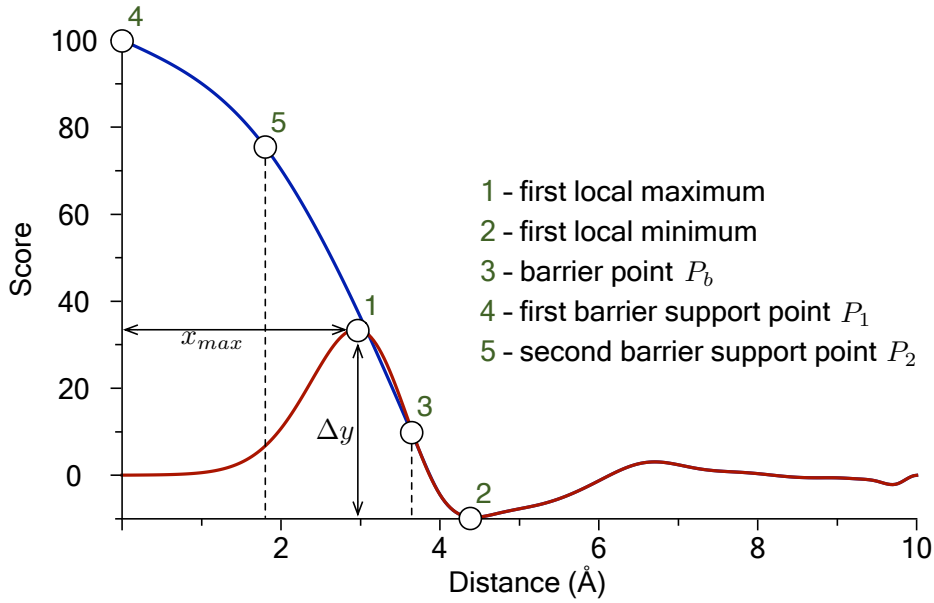


Figure 4.1 Schematic representation of the potential barrier reconstruction. Red - initial scoring function. Blue - the reconstructed potential barrier.

barriers with the rest of the scoring functions. In order to determine the parameters for the *barrier support* points, we exhaustively screenned a range of values and verified the results on the training set of protein complexes. Figure 4.1 schematically represents parameters used to reconstruct the potential barriers for the derived scoring functions.

### 4.3.2 Cross-Validation

One may highlight three parameters that influence the solution of the problem (4.8). The first parameter is the temperature factor $\sqrt{k_B T}$. This parameter controls the amplitude of the normal mode fluctuations (see Eq. 4.17). It affects the conformation of the generated decoys and hence the structure vectors $\mathbf{x}_{ij}$ extracted from these decoys. The second parameter is the standard deviation $\sigma$ of the Gaussian function (see Eq. 4.3), which also influences the structure vectors $\mathbf{x}_{ij}$. The third parameter is the set of regularization coefficients $C_{ij}$. The optimal values of these parameters are generally not known in advance. To estimate them, we used the *cross-validation* procedure. Namely, we screenned the values of these parameters in a certain range. Then, for each combination of the parameters we solved the optimization problem (4.8) on a reduced training set of 200 protein complex structures. After, we validated the obtained solutions on the other 644 protein complexes from the training set.

We screened the values of the temperature factor $\sqrt{k_B T}$ in $\{5, 10, 20, 40, 60\}$. The best

value for the Gaussian width parameter $\sigma$ was adopted from the previous study [30]. For the set of regularization parameters $C_{ij}$, we discriminate weights for the native and for the decoy structure vectors:

$$C_{ij}^{\text{native}} = C\frac{N_{\text{decoy}}}{N_{\text{total}}}$$

and

$$C_{ij}^{\text{decoy}} = C\frac{N_{\text{native}}}{N_{\text{total}}},$$

where $N_{\text{decoy}}$ and $N_{\text{native}}$ are the number of the decoy and the native structure vectors, respectively, and $N_{\text{total}} = N_{\text{decoy}} + N_{\text{native}}$ is the total number of the structure vectors. We choose parameters $C_{ij}$ to be different for the native and the decoy structure vectors of each complex because fewer native structure vectors should have larger weights. Thus the screening of the values of regularization parameters $C_{ij}$ is reduced to the screening of the values of the regularization constant $C$. We screened the values of the parameter $C$ from 1 to $10^9$ with the exponential step size of $\sqrt[3]{10}$. We found the optimal value for the temperature factor $\sqrt{k_BT}$, the standard deviation $\sigma$ and the regularization parameter $C$ to be equal to 10, 0.4, and $3.2 \times 10^4$, respectively. At the last step, we derived the final scoring functions using the complete training set with the optimal values of the parameters.

### 4.3.3 Rigid-Body Minimization

The scoring functions $\Upsilon^{kl}(r)$ (see Eq. (4.5)) are smooth by construction. This fact allows to use these functions for the structure optimization. More accurately, for a given $kl$-pair of atoms at a distance $r_{ij}$, the negative gradient $-\nabla\Upsilon^{kl}(r_{ij})$ could be regarded as the force with which one atom acts on the other atom. Thus, one may use the set of derived functions $\Upsilon^{kl}(r)$ to optimize a particular conformation of a protein complex until a local minimum is reached, provided $\nabla\Upsilon^{kl}(r_{ij}) = 0$ for each pair of atoms. Since special calibration is required to retain structure integrity of a complex, a more relevant structure optimization would be the *rigid-body* optimization, where instead of force minimization over each pair of atoms, one minimizes the net force and the net torque acting on each monomer. The rigid-body optimization with functions $\Upsilon^{kl}(r)$ could be useful in a local rigid-body minimization as a refinement step to process docking predictions. It was shown that such refinement could improve docking predictions dramatically [92]. In contrast to our scoring functions $\Upsilon^{kl}(r)$, most of modern statistical potentials are not differentiable [50, 121, 127, 152]. Thereby, to perform structure optimization with such potentials, one either uses a smooth interpolation of potentials, or employs various derivative-free optimization strategies, e.g. Nelder-Mead [98] or Powell [109] methods and their modifications, where the convergence rate

Table 4.1 The rigid-body minimization work-flow.

1. Set initial parameters for the structure optimization.

2. Compute the score $U_k$ of the current conformation and the descent direction $\mathbf{d}_k$ in the rigid-body space.

3. Find an appropriate step size $\alpha$ and make a step toward the descent direction: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$.

4. Repeat steps 2-3 until desired tolerance or maximum number of iterations is achieved.

5. Take the last computed score as the final score of the optimized conformation.

is much slower compared to first- or higher- order optimization strategies. Following this idea, we implemented the local rigid-body minimization protocol to explore whether such an optimization improves scoring capabilities of KSENIA. General work-flow for the local rigid-body minimization is listed in Table 4.1.

### 4.3.4 Normal Modes

Let us consider a system of $N$ particles with $3N$ degrees of freedom near the equilibrium state $\mathbf{x_0}$. The potential energy of the system can be approximated as a quadratic form:

$$U(x_1, x_2, \ldots, x_{3N}) = U(\mathbf{x_0}) + \frac{1}{2} \sum_{i=1}^{3N} \sum_{j=1}^{3N} F_{ij} x_i x_j, \tag{4.9}$$

where elements of the matrix of the quadratic form $F_{ij} = \left( \frac{\delta^2 U}{\delta x_i \delta x_j} \right)_{\mathbf{x_0}}$ are the force constants at the equilibrium state $\mathbf{x_0}$. There exist a different set of coordinates $y_i$, where both the kinetic $K$ and the potential $U$ energies have the *diagonal form* and thus the Newton's equations of motion are uncoupled. This means that the solution for the equations of motion for each coordinate can be obtained separately. These coordinates $y_i$ are called the *normal*

*coordinates*, and the corresponding energy terms have the following form:

$$\begin{aligned} U(y_1, y_2, \ldots, y_{3N}) = &\quad U(\mathbf{x_0}) + \frac{1}{2}\sum_{i=1}^{3N} \lambda_i y_i^2, \\ K(y_1, y_2, \ldots, y_{3N}) = &\quad \frac{1}{2}\sum_{i=1}^{3N} \dot{y}_i^2 \end{aligned} \tag{4.10}$$

The transition matrix between the two coordinate bases is obtained via diagonalization of matrix $\mathbf{M}^{-\frac{1}{2}}\mathbf{F}\mathbf{M}^{-\frac{1}{2}} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathbf{T}}$:

$$U - U(\mathbf{x_0}) \equiv \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{F}\mathbf{x} = \frac{1}{2}\mathbf{x}^{\mathbf{T}}\mathbf{M}^{\frac{1}{2}}\mathbf{L}\mathbf{D}\mathbf{L}^{\mathbf{T}}\mathbf{M}^{\frac{1}{2}}\mathbf{x} = \frac{1}{2}\mathbf{y}^{\mathbf{T}}\mathbf{D}\mathbf{y}, \tag{4.11}$$

where $\mathbf{M}$ is the diagonal mass matrix, i.e. $M_{ij} = m_i\delta_{ij}$. Thus, the connection between the two coordinate systems is given as a linear transformation

$$\mathbf{x} = \mathbf{M}^{-\frac{1}{2}}\mathbf{L}\mathbf{y} \tag{4.12}$$

Normal coordinates provide a convenient way to describe molecular fluctuations of a system near the equilibrium state. Particularly, the evolution of the system in the normal basis is the superposition of the independent harmonic oscillations along each normal coordinate $y_i$. Such oscillations are called normal modes [149] and are expressed as:

$$y_i(t) = A_i \cos(\omega_i t + \delta_i), \tag{4.13}$$

where $\omega_i \equiv \sqrt{D_{ii}}$ and $\delta_i$ correspond to the frequency and the phase of the $i$-th mode, respectively. The factor $A_i = \sqrt{2k_BT}/\omega_i$ is the amplitude of the fluctuation. Given the transition matrix $\mathbf{L}$ between the two bases (see Eq. (4.12)), oscillations in the Cartesian basis can be written as:

$$x_k(t) = L_{ki}(A_i \cos(\omega_i t + \delta_i))/\sqrt{m_k} \tag{4.14}$$

Thus, all atoms in a molecule for a given mode $i$ oscillate with the same frequency and phase. However, the amplitude of the fluctuation of the Cartesian coordinate $x_k$, corresponding to the oscillation of the mode $y_i$, is different for each coordinate $k$ and is defined by the $i$-th column of the transition matrix $\mathbf{L}$:

$$\langle x_k^2\rangle_i = L_{ki}^2 A_i^2 \langle \cos(\omega_i t + \delta_i)^2\rangle / m_k = \frac{1}{2m_k} L_{ki}^2 A_i^2 = L_{ki}^2 \frac{k_B T}{m_k \omega_i^2} \tag{4.15}$$

When all the modes are active, the amplitude of the fluctuation of the Cartesian coordinate

$x_k$ reads:

$$\langle x_k^2 \rangle = \frac{k_B T}{m_k} \sum_i \frac{L_{ki}^2}{\omega_i^2} \tag{4.16}$$

We use this theoretical framework to construct the training set of protein-protein complexes. A deeper discussion of normal modes analysis and its applications in structural biology can be found e.g. in [19, 21, 89, 145, 149].

### 4.3.5 Training Set

#### Native Complexes

We used the training database of 851 non-redundant protein-protein complex structures prepared by Huang and Zou [51]. This database contains protein-protein complexes extracted from the PDB [12] and includes 655 homodimers and 196 heterodimers. We updated three PDB structures from the original training database: 2Q33 supersedes 1N98, 2ZOY supersedes 1V7B, and 3KKJ supersedes 1YVV. The training database contains only crystal dimeric structures determined by X-ray crystallography at resolution better than 2.5 Å. Each chain of the dimeric structure has at least 10 amino acids, and the number of interacting residue pairs, as defined as having at least 1 heavy atom within 4.5 Å, is at least 30. Each protein-protein interface consists only of 20 standard types of amino acids. No homologous complexes were included in the training database. Two protein complexes were regarded as homologues if the sequence identity between receptor-receptor pairs and between ligand-ligand pairs was $> 70\%$. Finally, Huang and Zou [51] manually inspected the training database and left only those structures that had no artifacts of crystallization.

#### Near-native Decoys

To exclude any bias to computational methods and potentials for generation of putative binding poses, we construct our training set using structural information about only protein complexes in their native conformations. For the initial set of 844 native protein complexes (see Section 4.3.5) we generated near-native conformations, i.e. conformations within RMSD = 3 Å, for each native complex as follows. First, given the coordinate vector $\mathbf{X}^{\text{native}}$ of each monomer in a protein complex, we computed its ten lowest-frequency normal modes. Then, we formed fifteen near-native conformations for each monomer using the linear combinations of these modes :

$$\hat{\mathbf{X}} = \mathbf{X}^{\text{native}} + \sqrt{k_B T}\mathbf{M}^{-\frac{1}{2}} \sum_{i=1}^{10} r_i \frac{\mathbf{L}_i}{\omega_i}, \tag{4.17}$$

where $\sqrt{k_BT}$ is the temperature factor, $\mathbf{M}$ is the diagonal mass matrix, i.e. $M_{kl} = m_k\delta_{kl}$, $r_i$ is the random weight for each mode ranging from -1 to 1, $\mathbf{L}_i$ is the $i$-th column of the transition matrix between the Cartesian and the normal mode bases, and $\omega_i$ is the frequency of the $i$-th mode. The temperature factor $\sqrt{k_BT}$ affects the amplitude of the deformation, hence, too large temperatures cause a monomer to deform significantly breaking the covalent bonds. We tried several values of the temperature factor and found the optimal value of $\sqrt{k_BT}$ to be 10 $\text{kJ}^{\frac{1}{2}}$ (see Section 4.3.2). To ensure the absence of non-relevant conformations, we measured the RMSD between the native and the generated conformations. Indeed, the average RMSD is equal to 1.02 Å, which means that the deformations with the given temperature factor keep all generated conformations non-disrupted. At the last step, we combined conformations $\hat{\mathbf{X}}$ of two monomers representing one protein complex, resulted in $15 \times 15 = 225$ near-native conformations. To summarize, the composed training set to derive the scoring function contains 844 assemblies, where each assembly consists of one native protein complex and 225 generated near-native conformations.

We used the MMTK library [46] to perform the normal mode analysis for protein molecules and the OPLS-UA force-field [57] to compute the force constants (see Eq. (4.9) ). Since normal modes are defined for the equilibrium state of the system, we minimized each monomer of a dimer in a vacuum using 50 steps of the steepest descent algorithm with the relative energy tolerance of $1e-3$ and cut-off distance for all non-bonded interactions of 5 Å. We chose such a relatively small number of minimization steps in order to not significantly deform the X-Ray structure of a monomer. Indeed, the RMSD between the initial and the minimized monomer structures did not exceed 0.5 Å. Given each monomer near the equilibrium state, we used the Fourier subspace for the reduced-basis normal modes computations [45]. We picked up ten first low-frequency modes from the Fourier basis to generate different local deformations of the protein complexes. We should note that we excluded the first six modes that correspond to the rigid-body motion.

Finally, we want to stress that *all* generated conformations represent *near-native* protein structures. Indeed, we use directions along the slowest normal modes to locally deform the monomers, however, the orientations of the monomers with respect to each other are fixed. Since all the monomer conformations differ only slightly from the native monomers (the average RMSD is 1.02 Å), the interaction interfaces of all generated complexes undergo moderate changes keeping the major part of the native contacts. To conclude, we composed the training set based only on local information about the native interfaces and no other information was used. In the Results section we demonstrate the scoring function for protein-protein interactions derived using this training set.

### 4.3.6 Test Benchmarks

#### Hex Test Benchmark

For the first test, we constructed a rigid-body benchmark starting from the native structures in the training set. More precisely, to generate decoys we used the Hex rigid-body docking program [115, 116]. For the Hex input, we used polar Fourier shape expansions to polynomial order $N = 31$, the real-space angular search step of $7.5^\circ$, the radial search range of 40 Å with a translational step of 2.5 Å and the subsequent sub-step of 1.25 Å. We ran Hex for each native complex in the training set and clustered the docking solutions with a threshold of 8 Å. Top 200 docking predictions were added to the test benchmark in addition to the native complexes, resulting in $201 \times 844 = 169,644$ protein complexes. Finally, we evaluated the success rate of the Hex scoring function on the constructed benchmark according to the quality of the docking poses. Here we de ne the quality according to the value of RMSD of the backbone atoms of the ligand ($L_{\mathrm{RMSD}}$) after the receptors in the native and the decoy conformations have been optimally superimposed (see Table 1.1). To do so, we used the fast open-source RigidRMSD library [104] that computes RMSDs given spatial transforms of the docking poses.

#### Zdock Test Benchmark

For the second test benchmark we used the protein-protein docking benchmark v3.0 composed by Hwang *et al.*, which consists of 124 non-redundant protein-protein complexes [52]. Then, we employed Zdock v.3.0.1 rigid-body docking software [103], which uses a grid-based representation of two proteins and a three-dimensional fast Fourier transform to explore the search space of rigid-body docking positions. We used the bound conformation of each monomer in the benchmark for the Zdock input, randomly set initial protein orientations and used the default parameters for the docking predictions. Finally, we chose 2000 best generated rigid-body docking poses according to the Zdock v.3.0.1. scoring function for each complex. Thus, the second test benchmark consists of $124 \times 2,000 = 248,000$ protein complexes.

To evaluate the success rate of this scoring function on the constructed benchmark, we use the CAPRI criterion [56] for a correct prediction (Table 1.2).

#### Rosetta Test Benchmark

Gray *et al.* generated the Rosetta benchmark using 54 complexes from the protein-protein docking benchmark version 0.0 [24] in bothe the bound and the unbound conformations.

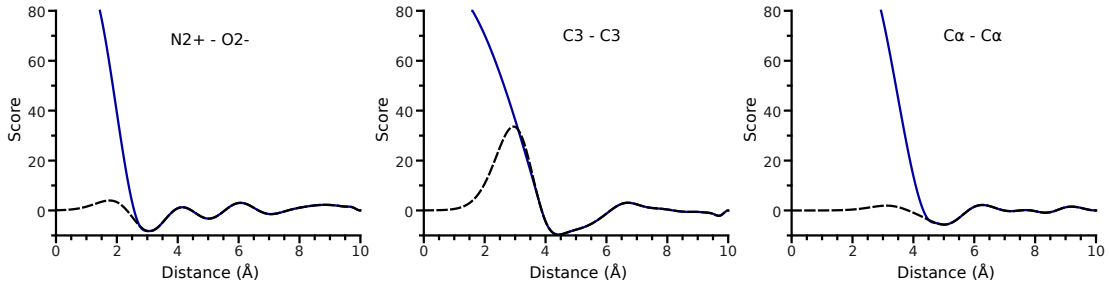


Figure 4.2 Examples of the derived distance-dependent scoring functions between atoms of types N2+ – O2- , C3 – C3 and $C_\alpha - C_\alpha$, respectively. Here, N2+ are guanidine nitrogens with two hydrogens, O2- are oxygens in carboxyl groups, C3 are aliphatic carbons bonded to carbons or hydrogens only and $C_\alpha$ are the backbone $C_\alpha$ atoms. Black, dashed: initially derived scoring functions without taking into account the absence of statistics at short distances. Blue, solid: redefined scoring functions that take into account the absence of statistics at short distances.

For each complex, the authors generated $1,000$ bound and $1,000$ unbound decoys following the flexible docking protocol, which is a part of the RosettaDock suite [41]. The first step in the protocol is the random translation and rotation of one of the proteins constituting the complex. Afterwards, the side chains are optimized simultaneously with the rigid body displacement of the protein. Finally, the full-atom minimization is performed to refine the conformation of the complex. We calculated the success rate of RosettaDock using the same quality criterion as in CAPRI [56] (Table 1.2). Both the bound and the unbound Rosetta benchmarks consist of $54 \times 1,000 = 54,000$ protein complexes.

## 4.4 Results

### 4.4.1 Scoring Functional

Figure 4.2 presents three derived scoring functions (dashed) for different atom pairs. As one can see, at short separation distances the scoring functions tend to zero. This is the artifact of the training set, and is mainly caused by the absence of observations of atom pairs at distances close to zero. However, we want our scoring functions to be able to penalize conformations in which steric clashes between the monomers are present. Thus, we re-define the scoring functions at short distances to form artificial potential barriers (see Section 4.3.1). The initial scoring functions along with the modified scoring functions are shown in Figure 4.2. We refer to the latter as to KSENIA, which stands for ***K****nowledge-based* ***S****coring function* ***E****mploying only* ***N****ative* ***I****nterfaces* .

The scoring functional $F$ (see Eq. (4.4)) of a particular protein complex $P$ is computed as the sum of separate scores for each pair of atoms within the cutoff distance $r_{\max}$. Thus, $F$, as a function of $3 \times (N_A + N_B)$ variables, where $N_A$ and $N_B$ are the numbers of atoms in molecules $A$ and $B$ respectively, is not identically zero only in the conformational volume where at least one pair of atoms is within $r_{\max}$ distance. Since KSENIA typically possesses several maxima and minima (see Fig. 4.2), $F$ is likely to be a rugged function in this volume [37]. However, we want to demonstrate that since our scoring functions were derived from the local deformations of the native conformations, the scoring functional $F$ is smooth at least in the neighborhood of the native conformation. To show this, we explored the behavior of the scoring functional $F$ in the four-dimensional manifold of the $3 \times (N_A + N_B)$ conformational space. Namely, given two monomers, one of which is fixed, we consider four coordinates corresponding to the rigid-body degrees of freedom: the distance $d$ between the centers of mass of the two monomers, the rotation of the free molecule about the axis connecting the centers of mass by an angle $\alpha$ , and two rotations about two other orthogonal axes by angles $\beta$ and $\gamma$. Then, starting from the native conformation of the complex $(d_0, \alpha_0, \beta_0, \gamma_0)$, we calculate partial derivatives in the vicinity of this conformation. More precisely, we sample the first partial derivative $\frac{\delta F(d,\alpha,\beta,\gamma)}{\delta e}$ at points $\{e_0 \pm \varepsilon, e_0 \pm 2\varepsilon, e_0 \pm 3\varepsilon, \dots\}$, where $e \in \{d, \alpha, \beta, \gamma\}$, and $\varepsilon$ is a sufficiently small positive value. At the point where the partial derivative changes its sign, we can not expect a gradient-based local minimization algorithm to find the nearest local minimum to the point $(d_0, \alpha_0, \beta_0, \gamma_0)$. Thus, one can characterize the smoothness of the scoring functional $F$ at the point $(d_0, \alpha_0, \beta_0, \gamma_0)$ by four intervals $(e_0 - m\varepsilon\ ,\ e_0 + n\varepsilon)$, where the partial derivative is a constant-sign function. Figure 4.3 shows the distribution of such interval lengths over the native conformations in the training set. The most probable size of the smooth region around the native conformation is 2.2 Å, 0.42 rad, 0.22 rad, 0.22 rad in four degrees of freedom, respectively. Practically, it means that the rigid-body minimization, started from an arbitrary point within this region, is expected to optimize the conformation corresponding to this point toward the conformation corresponding to the local minimum of this region, assuming that $F$ is convex in the neighborhood of the native conformation.

Finally, it remains to prove that the point representing the native conformation in the four-dimensional manifold lies close to the local minimum. To demonstrate this, we measured the RMSD between the native conformation and the conformation obtained after the rigid-body minimization with the KSENIA potential starting from the native conformation. Figure 4.4 shows the distribution of such RMSDs in the training set. As it could be seen, the minimized and native structures are very similar and the corresponding RMSD does not exceed 2 Å. Moreover, the most probable RMSD between the two conformations is 0.1 Å.

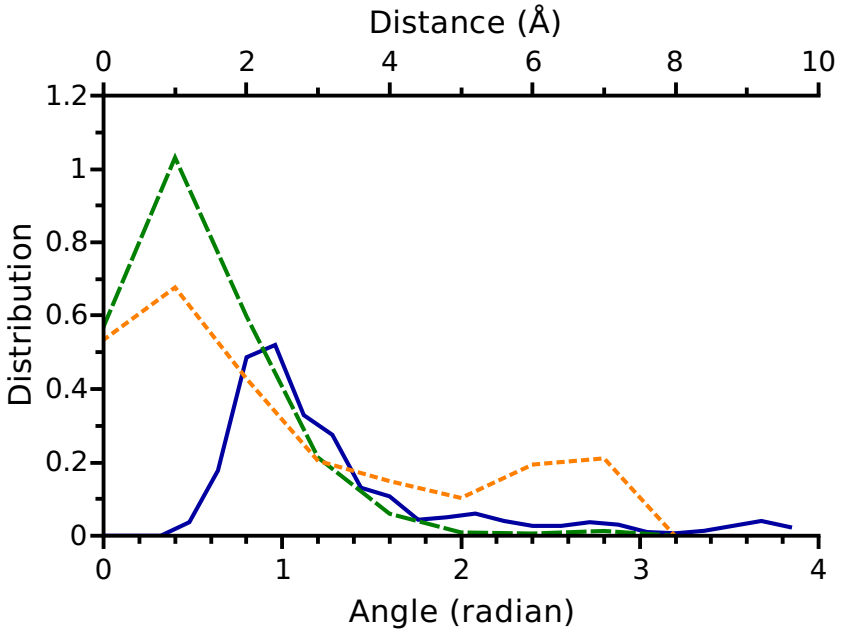


Figure 4.3 Distribution of the interval lengths in the four-dimensional manifold where the partial derivatives of the scoring functional are the constant-sign functions. These distributions are computed using the native structures in the training set. Blue, solid: interval length for the $d$-coordinate, which is the distance between the centers of mass of two monomers. Green, dashed: interval length for the $\alpha$-coordinate, which is the angle of rotation of the ligand about the axis connecting the centers of mass. Orange, dotted: interval length for the $\beta$- and $\gamma$- coordinates, which are the angles of rotation about two other orthogonal axes.

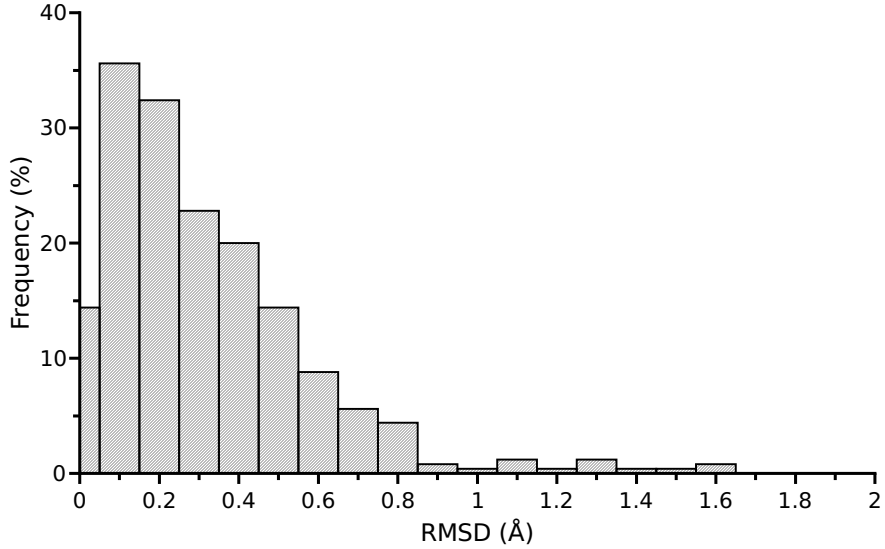


Figure 4.4 Histogram representing the distributions of the RMSDs between the native and minimized conformations in the training set using the rigid-body minimisation protocol.

To summarize, we demonstrated that the scoring functional $F$ is a smooth function in the vicinity of the native conformation. Hence, the rigid-body minimization is expected to improve predictions if started at an arbitrary point in this vicinity. Below, we provide numerical experiments that demonstrate the practical importance of the rigid-body minimization with KSENIA.

### 4.4.2 Performance on the Test Benchmarks

The aim of any scoring function is to differentiate the native and near-native conformations of protein complexes from the non-native ones. In this section we demonstrate that observing only the native protein complexes is sufficient to build a powerful and well-discriminative knowledge-based scoring function. Using four different protein-protein benchmarks described in Section 4.3.6, we evaluate the success rate of our method, which is defined as the percentage of protein complexes for which docking predictions with quality-one, -two or -three are ranked at the top positions. We also compare our method with the widely-used scoring functions of Hex [115], Zdock [103], and Rosetta [41].

#### Hex Test Benchmark

In the first test, we used the Hex test benchmark (see Section 4.3.6). Although the training set and this benchmark share the same native structures, their decoys are very different.

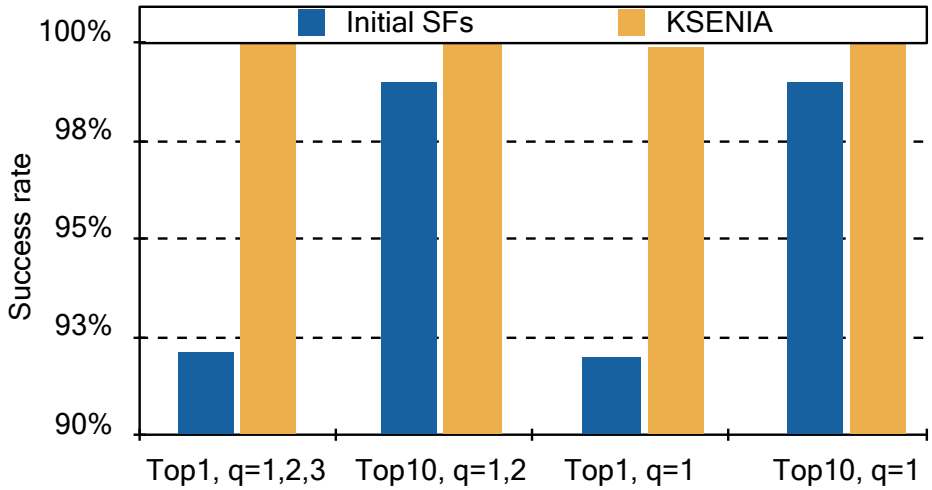


Figure 4.5 Performance of the scoring functions on the Hex test benchmark. Success rates of the initial scoring functions (Initial SFs) are depicted with the blue rectangles. Success rates of KSENIA are depicted with the yellow rectangles. Top$N$ value is defined as the percentage of protein complexes for which at least one of the docking prediction with the corresponding quality $q$ is present within the first $N$ docking poses. The quality of predictions $q$ is evaluated according to the value of $L_{\mathrm{RMSD}}$ (see Table 1.1).

More precisely, for the training, we generated *local deformations* at the protein-protein interfaces for all native complexes using directions along the low-frequency normal modes. On the other hand, to generate decoys for the test benchmark, we performed the exhaustive search in the six-dimensional space of rigid-body motions. Consequently, many different interfaces for each native complex are present. Furthermore, owing to the clustering of spatially close docking predictions, there are no similar interfaces in the test benchmark. Thus, the goal of the first test is to demonstrate that employing only local information about the native interfaces is sufficient to derive a well-discriminative scoring function. We ranked all docking poses in the training set according to the values of the initial scoring functions and the values of KSENIA. Figure 4.5 presents the corresponding success rates for the top predictions. Clearly, the derived scoring functions predict the native interfaces very well, providing the success rates of more than 90% for the top one predictions. To explore if our scoring functions can distinguish correct interfaces (generated by Hex with quality-one, -two or -three) from the non-native ones, we removed the native structures from the test benchmark, leaving only predictions with non-zero rotational part of the spatial transform. We will refer to the obtained set as to the *reduced* Hex test benchmark. Figure 4.6 shows recomputed success rates for the top predictions (solid rectangles). In this figure, we also list the maximum success rates of the scoring functions (hollow rectangles) as the percentage of protein complexes for which Hex could predict poses of the corresponding quality. From

Figure 4.6 one can see that the derived scoring functions provide a similar success rate as the Hex scoring function, which is solely based on the shape-complementarity term. However, the initial scoring functions slightly out-perform KSENIA on the reduced Hex test benchmark. Presumably this is because we lose some information when re-defining potentials at short distances (see Section 4.3.1). Nonetheless, KSENIA is dedicated to be used with the local rigid-body minimization for the refinement of the docking predictions. Thereby, at the next step we used the rigid-body minimization protocol (see Section 4.3.3 and Table 4.1 ) to optimize the docking poses. Then, we ranked the optimized docking predictions according to the values of KSENIA and re-evaluated the success rates (Figure 4.6, green solid rectangles). We found that the rigid-body minimization dramatically improves the scoring results. In particular, the rigid-body minimization increased the total number of quality-one poses, rising the maximum success rate from 28% to 66%. Moreover, the corresponding success rates are more than twice better compared to both the success rates of Hex and the success rates of scoring without the refinement procedure. To summarize, we demonstrated that employing structural information of only native interfaces, it is possible to distinguish near-native conformations of protein complexes from the non-native decoys. We have also shown that it is possible to refine docking predictions using a smooth knowledge-based statistical scoring function with a rigid-body minimization algorithm, which improves the quality of the predictions and the overall performance of the scoring method. Below, we further investigate the capability of our approach on more complicated test benchmarks.

### Zdock Test Benchmark

For the Zdock benchmark set (see Section 4.3.6 ) we applied the rigid-body minimisation protocol with KSENIA, as in the previous section, ranked the poses and compared the success rates against Zdock v.3.0.1 scoring function, which includes the shape-complementarity term, the electrostatic term and the desolvation term. Figure 4.7 shows results obtained on this benchmark. Our approach shows around three times better success rate for the top one quality-one, -two or -three predictions. We should note, however, that for eight complexes in the benchmark, the rigid-body minimization deteriorated several quality-one predictions to quality -two or -three. Thus, the maximum number for the top one quality-one predictions is reduced from 97% to 91%. Nonetheless, our method demonstrates around seven times higher success rate for the top one predictions with the highest quality compared to the Zdock v.3.0.1 scoring function. We should note that we did not verify the performance of KSENIA on the protein-protein unbound benchmark [52]. After the rigid-body docking applied to the monomers in the unbound conformations, side-chains of the interface residues are, generally, in non-optimal conformations, which might be crucial for KSENIA.

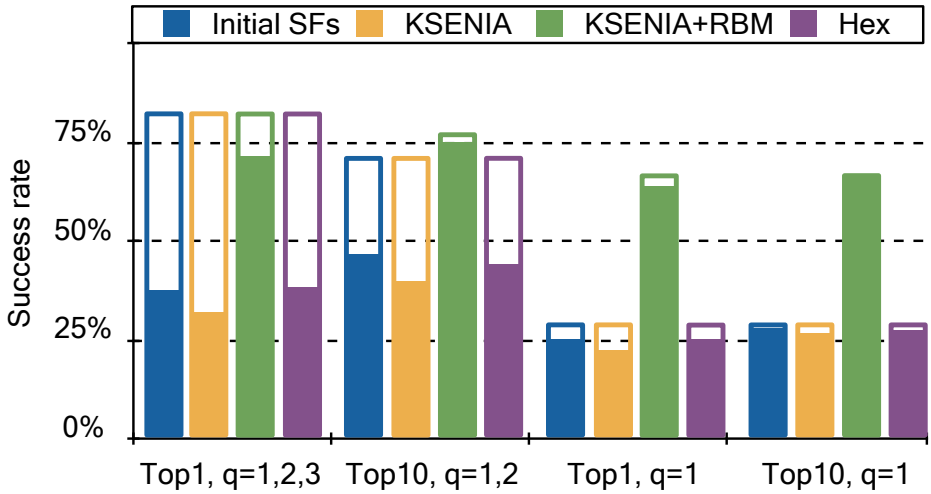


Figure 4.6 Performance of the scoring functions on the reduced Hex test benchmark. Success rates of the initial scoring functions (Initial SFs) are depicted with the solid blue rectangles. Success rates of KSENIA are depicted with the solid yellow rectangles. Success rates of KSENIA along with the rigid-body minimization (KSENIA+RBM) are depicted with the solid green rectangles. Success rates of the Hex scoring function are depicted with the solid purple rectangles. Hollow rectangles of the corresponding color represent the maximum achievable success rates. Top$N$ value is defined as the percentage of protein complexes for which at least one of the docking prediction with the corresponding quality $q$ is present within the first $N$ docking poses. The quality of predictions $q$ is evaluated according to the value of $L_{\mathrm{RMSD}}$ (see Table 1.1).

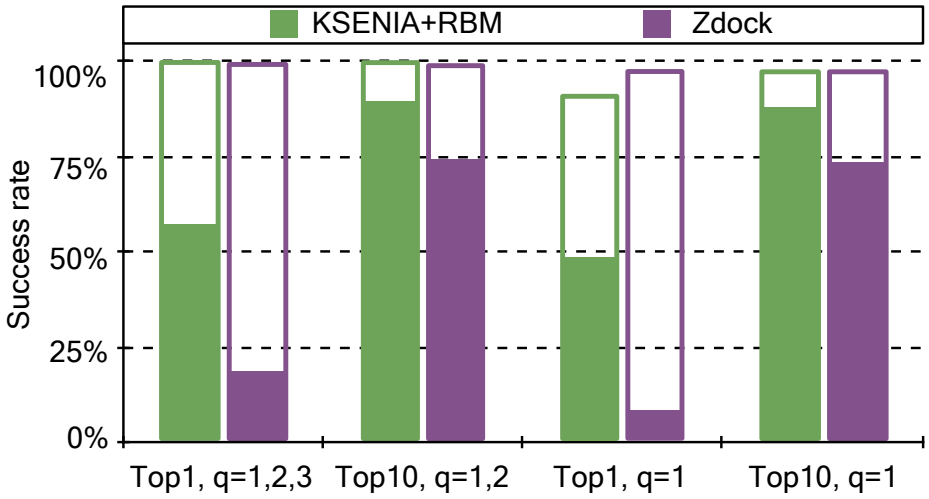


Figure 4.7 Performance of the scoring functions on the Zdock test benchmark. Success rates of KSENIA along with the rigid-body minimization (KSENIA+RBM) are depicted with the solid green rectangles. Success rates of the Zdock scoring function are depicted with the solid purple rectangles. Hollow rectangles of the corresponding color represent the maximum achievable success rates. Top$N$ value is defined as the percentage of protein complexes for which at least one of the docking prediction with the corresponding quality $q$ is present within the first $N$ docking poses. The quality of predictions $q$ is evaluated according to the CAPRI criterion (see Table 1.2).

Instead, we verified the performance of KSENIA on the Rosetta bound and unbound test benchmarks, where side-chain conformations are optimized.

### Rosetta Test Benchmark

Comparison of the performance of the Rosetta's scoring function against our rigid-body minimization with KSENIA is presented in Figure 4.8 for both the bound and the unbound benchmarks. As it could be seen, although Rosetta itself performs slightly better, our approach still demonstrates very good results despite the complexity of these benchmarks. Indeed, the native contacts for all the complexes in the benchmark are disturbed owing to the side-chain re-packing or homologous replacement, for example. In addition, our scoring method does not take into consideration the individual scores of the monomers. In particular, it does not penalize rare rotameric states of the side-chains, which are present in the benchmark. Nonetheless, using only distance distributions between the atoms in different monomers at their native and near-native states, our scoring function is capable to rank quality-one poses at the top position for around 60 % of cases for the Rosetta bound benchmark, and to rank quality-one, -two or -three poses at the top position for around 45 % of cases for the Rosetta unbound benchmark.

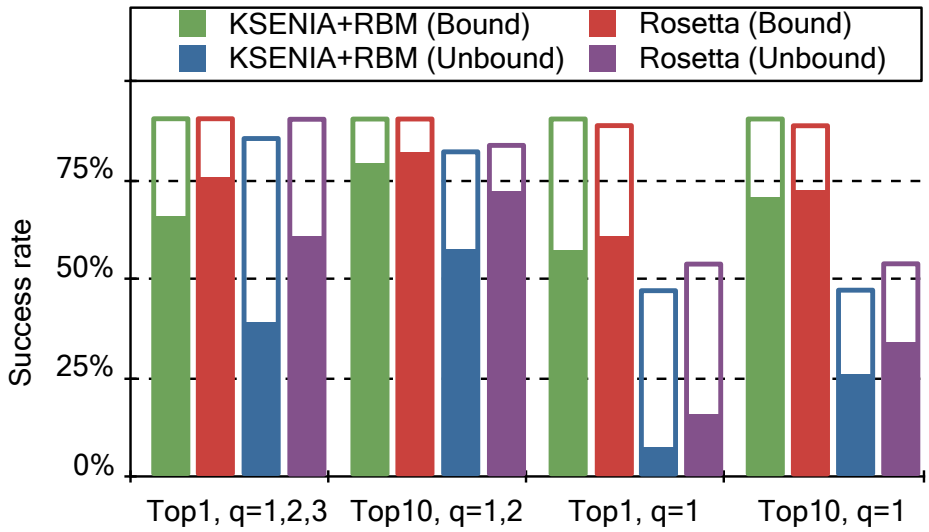


Figure 4.8 Performance of the scoring functions on the Rosetta bound and unbound test benchmarks. Success rates of KSENIA along with the rigid-body minimization (KSENIA+RBM) are depicted with the solid green and the solid blue rectangles for the Rosetta bound and unbound test benchmarks, respectively. Success rates of the Rosetta scoring function are depicted with the solid red and the solid purple ractangles for the Rosetta bound and unbound test benchmarks, respectively. Hollow rectangles of the corresponding color represent the maximum achievable success rates. Top$N$ value is defined as the percentage of protein complexes for which at least one of the docking prediction with the corresponding quality $q$ is present within the first $N$ docking poses. The quality of predictions $q$ is evaluated according to the CAPRI criterion (see Table 1.2).

Table 4.2 Scores for the native and one of the decoy structures before and after the rigid-body minimization.

| 1ZC6 | Score | Score after the rigid-body minimization |
|---|---|---|
| $U_{\text{decoy}}$ | -1594.740 | -3036.307 (rank 1) |
| $U_{\text{native}}$ | -1810.758 (rank 1) | -2144.868 |

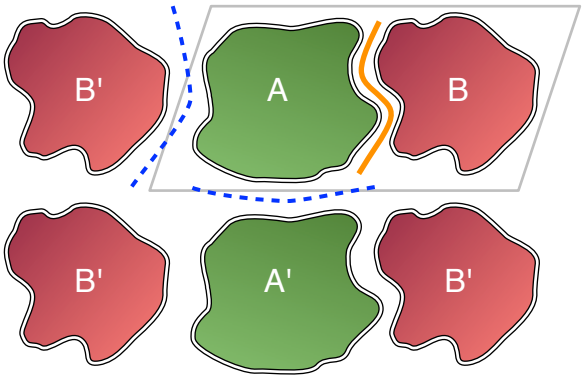


Figure 4.9 Schematic representation of the native interface (orange, solid) and crystal contacts (blue, dashed). The unit cell is depicted as the gray parallelogram encompassing monomers A and B, which form the native interface.

### 4.4.3 Crystallographic Symmetry Mates as Docking Predictions

We observed that in several cases non-native decoys replace near-native predictions at the top positions after the rigid-body minimization applied. As the result, the success rate becomes less than it could be, since the near-native predictions get a lower rank. For example, Table 4.2 lists scores before and after the rigid-body minimization applied to the protein complex 1ZC6 from the Hex test benchmark. In terms of the ligand-RMSD, the decoy structure significantly differs from the native one: $L_{\text{RMSD}} > 60$ Å. However, we found that the interface formed by the decoy monomers is similar to the one of the crystal-packing interfaces that are observed in the crystal structure. Typically, only one of the interfaces presented in the crystal is considered to be the native interface, and other crystal-packing interfaces or crystal contacts are considered to be the artifacts of crystallization (Fig. 4.9). However, distinguishing between the native interface and the crystal contacts is a challenging problem, since both are formed following the same physical principles [68, 73]. For the case of homodimer 1ZC6, $L_{\text{RMSD}}$ between the decoy and the complex forming the crystal contact is about 2.8 Å. We found these observations to be the additional evidence of the prediction capability of KSENIA.

### 4.4.4 Discussion

Reference state-based statistical methods require a large set of *false-positive* examples of protein complexes, i.e. non-native conformations, in order to compute the reference state. Linear and quadratic programming approaches train the scoring function also involving a set of generated *false-positive* examples in order to construct the system of inequalities (4.1). It is a common practice in protein-protein docking to select as *false-positive* examples those decoys that possess the best score according to some well-accepted scoring function [25, 51, 138]. On the contrary, we have selected *false-negative* examples purely based on the structure of protein complexes in their native conformations. More precisely, our decoy sets were generated in such a way that the average RMSD between the corresponding monomers in the decoys and in the native structures is about 1 Å, keeping the relative orientation of the monomers fixed. Nonetheless, despite our training set does not contain non-native conformations with large RMSDs with respect to the native structures, we are able to reconstruct the atom-atom distance-dependent scoring functions (see Eq. 4.5). As we have shown above, the obtained potentials demonstrate surprisingly good results on four protein-protein docking benchmarks. We would like to emphasize that all the benchmarks mostly consist of non-native decoys that have large RMSDs with respect to the native structures. Thus, our results strongly suggest that the native protein complexes themselves contain all necessary structural information to build well-discriminative potentials that recognize native and near-native protein-protein conformations.

Regarding the disadvantages of the proposed methodology, i.e. derivation of the KSENIA potential, we can point out two aspects. First, current statistic observations do not take into account conformations of individual monomers. This means that, in principle, we can imagine a situation when two very unrealistic structures of two monomers (all atomic coordinates inside each monomer are the same, for example) result in a good score of the complex. To circumvent this problem, one may either collect extra geometric information, such as triplet, quadruplet, etc. distributions of atoms in the complex, or additionally score individual monomers. Second, in our training set there are no statistics at short separation distances between the monomers inside a complex. Thus, as a result, we need to define potential barriers at short distances for the proper behaviour of the obtained scoring functions.

We would also like to stress that the derived KSENIA potential has no bias toward a method to generate docking predictions. This is because for the construction of the training set we did not use any standard docking prediction method such as Zdock, Hex, etc. Thus, the rigid-body minimization is very important for the success of the proposed scoring methodology. Namely, the minimization is required to resolve steric clashes, which often appear in docking predictions produced by various methods. For example, Zdock and Hex

use a soft shape complementarity potential, which permits moderate overlap between the monomers in a complex. Generally, we believe that structure optimization should be the inevitable step of a general scoring procedure when one has no information about docking predictions to score.

Our method does not, in principle, require external packages, potentials, or algorithms neither to generate the training set, nor to formulate and solve the optimization problem. In the present study, to generate the local deformations, we computed low-frequency normal modes using the MMTK package with a united-atom force-field [46]. However, normal modes can be computed in a simpler way using, e.g. the elastic-network model [137], the gaussian network model [6], the rotation-translation of blocks method [133], etc. Thus, methodology presented in this paper can be easily adapted to the recognition of other types of molecular interactions, such as protein-ligand, protein-RNA, etc., provided that the atom types assignment is modified appropriately.

## 4.5 Conclusions

Present study demonstrates that knowledge of only native protein-protein interfaces is sufficient to construct well-discriminative predictive models for the selection of binding candidates. Namely, we introduced a new scoring method that comprises a knowledge-based scoring function called KSENIA deduced from the structural information about the native interfaces of 844 crystallographic protein-protein complexes. The knowledge-based potential relies on the information obtained thanks to the deformations of these interfaces computed along the low-frequency normal modes. As a result, in contrast to existing scoring functions, our potential does not require neither the computation of the reference state nor the ensemble of non-native complexes. Thus, it has no bias toward a method to generate putative binding poses. Moreover, KSENIA is smooth by construction, which allows to use it along with the gradient-based rigid-body minimization. Particularly, we showed that the rigid-body optimization of the docking poses improves the scoring stage of molecular docking. Using several test benchmarks we demonstrated that our method out-performs the Hex scoring function, which is based on the shape complementarity between the monomers in a complex, and the Zdock scoring function, which also includes the electrostatic and desolvation terms. We found remarkable that the native protein complexes themselves contain all necessary information to derive a successful and well-discriminative scoring function. Although our method performs slightly worse on the Rosetta test benchmark compared to the more sophisticated RosettaDock scoring function, we believe that further improvements of KSENIA, e.g. accounting for the integrity of monomers, rotamer optimization, etc., will

eliminate this disadvantage.

Methodology presented in this paper can be easily adapted to the recognition of other types of molecular interactions, such as protein-ligand, protein-RNA, etc. We will make KSENIA publicly available as a part of SAMSON software platform developed in our group at http://nano-d.inrialpes.fr/software.

# Chapter 5

# CARBON: Controlled-Advancement Rigid-Body Optimization for Nanosystems

## 5.1 Introduction

Most modern docking algorithms are dedicated to predicting the bound state of a molecular complex from the structure of its unbound subunits. Given an initial set of binding candidates, various refinement algorithms are involved to take into account the flexibility of molecular complexes [145, 151] or to get rid of docking artefacts, e.g. overlaps between subunits of a molecular complex. To address the latter problem, one possibility is to continuously minimize the energy of the complex with respect to rigid-body transformations [16]. The rigid-body motion formalism aims at characterizing the location of rigid objects, and has obvious uses in the description of robot kinematics [42, 79, 140].

In biological applications, one of the methods commonly used to perform a rigid-body minimization is to apply rigidity constraints to an all-atom optimization, as described for example in the original CHARMM paper [20]. Another approach consists in computing generalized forces that act on molecules considered as rigid bodies, and in solving differential equations to update generalized velocities and molecular coordinates [26]. Recently, Mirzaei et al. [92] described a fast rigid-body minimization algorithm for refinement of docking predictions. The authors used local parametrization of rigid transformations $SO(3) \times R^3$ with exponential coordinates and defined rigid-body minimization as an optimization problem on the $R^6$ Euclidean space. The optimization problem was solved with the limited-memory BFGS algorithm (L-BFGS) [80]. Their method was adjusted and ap-

plied to refine docking predictions produced by the Piper software [72] with the CHARMM force-field. The authors reported that the use of their rigid-body formalism results in a one-order of magnitude speedup, when compared to an all-atom optimization with constraints. However, the authors also noticed several shortcomings, such as unstable behavior of the method when the monomers in the molecular complex approach too close to each other.

In this study, we present a fast rigid-body minimization approach which uses the net generalized force as a descent direction in a six-dimensional manifold. To circumvent the problem of incorrect step sizes for rotational and translational movements of molecular complexes, we introduce the concept of *controlled advancement*. Precisely, we use a recently introduced expression for the root-mean-square deviation (RMSD) between two molecular complexes [104] to control the minimum and maximum distances that rigid bodies are allowed to travel when performing optimization in a given direction. We demonstrate the efficiency of our approach in combination with classical empirical potentials, e.g. the CHARMM force-field, as well as with knowledge-based scoring functions, for which there is currently growing interest in virtual screening. Furthermore, using the knowledge-based scoring function previously derived in our lab, we show that refinement with our rigid-body minimization method dramatically improves results of the scoring stage of the docking pipeline. We compare our method with the state-of-the-art rigid-body minimization approach of Mirzaei et al. [92] on a set of protein-protein complexes. Finally, we show that the presented rigid-body minimization algorithm is able to resolve soft, moderate and large steric clashes in molecular complexes.

## 5.2 Theoretical Foundation

### 5.2.1 Rigid-Body motion representation

In this paper, we represent a rigid-body motion as a pair of operators, the rotation and translation operators, applied to the rigid-body of interest to change its position and orientation in space. First, the rotation operator is applied, and then it is followed by the translation operator. Whereas in most cases the translation operator is simply expressed as a 3-vector in Euclidean space, the rotation operator may be written in several ways. For example, it can be represented via a rotation matrix, Euler's angles, a quaternion or an axis-angle representation. Although all these representations may be considered equivalent, some are more numerically efficient than the others. Here, we use quaternions (see Section 3.2) since we found them more beneficial and convenient compared to the other representations [104].

### 5.2.2 Rigid-Body Energy Minimization

Given two monomers $A$ and $B$ of a molecular complex $AB$ and the potential energy function $U$, the problem of the rigid-body optimization is to find the rigid transformations for $A$ and $B$ that minimize the interaction energy $U_{AB}$ between them. To solve this problem, one often considers the *local* rigid-body minimization, which is the search of the deepest minimum of the potential energy function in the set of rigid transformations corresponding to local changes of the structure of the molecular complex $AB$. These local changes are typically characterized in terms of the RMSD from the initial conformation or relative orientation of monomers. The rigid transformations reduce the dimensionality of the conformational space down to six degrees of freedom corresponding to the mutual translation and rotation of the two rigid bodies. Thus, the rigid-body minimization could be expressed as an optimization problem with respect to the rotation and translation operators. However, regarding the rigid-body minimization of molecular complexes, one may encounter several pitfalls mainly related to the rugged shape of the potential energy landscape. For example, if steric clashes are present in the conformation of a molecular complex, the magnitude of the gradient $\mid \nabla U_{AB} \mid$ could be enormously large, typically resulting in very large moves of monomers in the complex with respect to each other. On the contrary, too small magnitudes of the gradient result in irrelevantly small moves of the monomers. Below, we describe a novel fast approach for the rigid-body minimization, which abates the influence of the above-mentioned drawbacks.

#### Force and position update

We use the rigid-body dynamics formalism to describe forces and torques acting on a rigid body [7]. Precisely, we view the rigid-body optimization problem as the calculation of *quasi-static* trajectories of rigid bodies influenced by a force-field, i.e. trajectories where rigid-body velocities are zeroed at the end of each time step[1], and rigid bodies follow the *inverse-inertia-weighted* energy gradient.

Given the potential energy function $U(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N)$, the force $\mathbf{F}_j^A$ acting on the $j$-th particle of rigid body $A$ is computed as the negative gradient of $U$. The net force acting on $A$ is then given as the superposition of forces acting on each particle: $\mathbf{F}^A = \sum_j \mathbf{F}_j^A$. The forces $\mathbf{F}^A$ and $\mathbf{F}^B = -\mathbf{F}^A$ provide translation directions for the monomers $A$ and $B$, respectively. Assuming a point $\mathbf{r}$ with mass $m$ has zero velocity at the beginning of the time step, and has

[1] Strictly speaking, quasi-static trajectories demand that rigid-body velocities are zero at all time, but we use a *discretized* point of view that is more natural in the context of rigid-body simulation.

a constant acceleration **a** for a duration $\tau$, its quasi-static displacement $\Delta\mathbf{r}$ is

$$\Delta\mathbf{r} = \frac{\mathbf{a}\cdot\tau^2}{2} = \frac{|\,\mathbf{F}\,|\cdot\tau^2}{2m}, \tag{5.1}$$

where **F** is the force applied to point **r**. As a result, the translation operator $\hat{\mathbf{T}}$ applied on a rigid body corresponds to the translation vector **T**

$$\mathbf{T} = \mathbf{F}\cdot\frac{\tau^2}{2M} \tag{5.2}$$

where $M$ is the total mass of the rigid body. Because the translation operator takes into account the mass of the rigid body, a heavier monomer is displaced less compared to a lighter one (it is a more inert rigid body).

Until now, we did not consider that the force acting on a particular point of the rigid body sets a spin, i.e. rotational motion, on it. To characterize this motion, the torque $\mathbf{G}^A_j$ resulting from the action of the force $\mathbf{F}^A{}_j$ on the $j$-th particle of $A$ is computed as

$$\mathbf{G}^A{}_j = (\mathbf{r}^A{}_j - \mathbf{r}^A{}_c)\times\mathbf{F}^A{}_j, \tag{5.3}$$

where $\mathbf{r}^A_j$ is the position of the $j$-th particle of $A$ and $\mathbf{r}^A_c$ is the center of mass of $A$. Similarly to the net force, the net torque $\mathbf{G}^A$ acting on $A$ is computed as follows

$$\mathbf{G} = \sum_j \mathbf{G}^A{}_j = \sum_j (\mathbf{r}^A{}_j - \mathbf{r}^A{}_c)\times\mathbf{F}^A{}_j \tag{5.4}$$

The net torque depends on the position of the particles relative to the center of mass and, in contrast to the net force, does convey the information about the distribution of forces acting on the rigid body. Assuming the rigid-body has zero angular velocity at the beginning of the time step and a constant torque applied to it, we compute the angular velocity at the end of the time step as follows

$$\mathbf{w} = \mathbf{I}^{-1}\cdot\mathbf{G}\cdot\tau, \tag{5.5}$$

where **I** is the inertia tensor of the rigid body,

$$\mathbf{I} = \begin{pmatrix} \sum m_i(y_i^2+z_i^2) & -\sum m_i x_i y_i & -\sum m_i x_i z_i \\ -\sum m_i x_i y_i & \sum m_i(x_i^2+z_i^2) & -\sum m_i y_i z_i \\ -\sum m_i x_i z_i & -\sum m_i y_i z_i & \sum m_i(x_i^2+y_i^2) \end{pmatrix}, \tag{5.6}$$

and $m_i$ with $\{x_i, y_i, z_i\}$ are the mass and position of the $i$-th particle, respectively. Note, that once the inertia tensor is computed in the reference frame ($\mathbf{I}_{ref}$), the inertia tensor in another

frame can be expressed as

$$\mathbf{I} = R \cdot \mathbf{I}_{ref} \cdot R^T, \tag{5.7}$$

where the rotation matrix $R$ corresponds to the transition between the two frames. Finally, given the angular velocity $\mathbf{w}$, we update the rotational quaternion $\hat{Q}$ according to

$$\dot{\hat{Q}} = \frac{1}{2}[0, \mathbf{w}]\hat{Q} \tag{5.8}$$

As one can see from Eqs. (5.5), (5.6), and (5.8), the obtained rotation quaternion $\hat{Q}$ involves the mass distribution of the rigid body, which influences the angle of rotation similarly to how the mass of a rigid body $M$ influences the translation step (Eq. (5.2)).

Finally, given the translation vector $\mathbf{T}$ and the rotation quaternion $\hat{Q} = [s, \mathbf{q}]$, the new position of the rigid-body is expressed as

$$\mathbf{r}^{\text{new}} = \mathbf{r}^{\text{old}} + 2\mathbf{q} \times (\mathbf{q} \times \mathbf{r}^{\text{old}} + s\mathbf{r}^{\text{old}}) + \mathbf{T} \tag{5.9}$$

We will refer to the couple of forces $(\mathbf{F}, \mathbf{G})$ as to the generalized force acting on a rigid body and will use it as the descent direction $\mathbf{d}$ in the six-dimensional Euclidean space. To simplify the notation, we will also denote displacement of a rigid body upon the generalized force as $\mathbf{x}^{\text{new}} = \mathbf{x}^{\text{old}} + \tau \cdot \mathbf{d}$ and refer to the time step $\tau$ as to the step size for the descent direction $\mathbf{d}$.

### Controlled advancement and acceptance criterion

Given the descent direction $\mathbf{d}$, the algorithm to determine an appropriate step size along this direction is required. In order to do this, we introduce the *advancement region* $(\tau_{\min}, \tau_{\max})$, that is the interval of $\tau$ that correspond to *allowed displacements* of a molecule, and we define the allowed displacements as a set of rigid-body displacements with RMSDs to the original position within a range $(\text{RMSD}_{\min}, \text{RMSD}_{\max})$. To express the step size $\tau$ via the RMSD upon a rigid-body transformation, we use the following relation [104]

$$\text{RMSD}^2 = \frac{4}{M}\mathbf{q}^T\mathbf{I}\mathbf{q} + \mathbf{T}^2. \tag{5.10}$$

where $M$ is the mass of a monomer, $\mathbf{q}$ is the vector part of a rotation quaternion, $\mathbf{I}$ is the inertia tensor and $\mathbf{T}$ is the translation vector. Using Eqs (5.1), (5.5), and (5.8) one obtains

$$\text{RMSD}^2 = \frac{\tau^4}{M(1 + \frac{\tau^4(\mathbf{I}^{-1}\mathbf{G})^2}{4})}\mathbf{G}^T\mathbf{I}^{-1}\mathbf{G} + \frac{\tau^4\mathbf{F}^2}{4M^2} \tag{5.11}$$

Using this equation, we can compute the bounds on the advancement region $\tau_{\min}$ and $\tau_{\max}$ as a function of $\mathrm{RMSD}_{\min}$ and $\mathrm{RMSD}_{\max}$, respectively. To do so, we need to solve a quadratic equation with respect to $\tau^4$. However, to optimize a molecular structure with a steric clash, we may consider only the translational RMSD. Thus, a simpler option to define the bounds on $(\tau_{\min}, \tau_{\max})$ is to compute step sizes $\tau$ corresponding to the minimum and the maximum translation magnitudes $T$ using Eq. (5.1),

$$\tau = \sqrt{\frac{2MT}{|\,\mathbf{F}\,|}}. \tag{5.12}$$

In this study with use the minimum translation of $T_{\min} = 0.001$ Å and the maximum translation of $T_{\max} = 3$ Å. We should note that the value of $T_{\min}$ is an order of magnitude smaller compared to the accuracy of the PDB molecular format, thus it is appropriate for our optimization method and we do not need to make it smaller. The value of $T_{\max}$ guarantees that the linear step-size search starts with the initial value of $\tau$ that corresponds to the RMSD between monomer' conformations of greater than 3 Å.

Given this and concepts introduced in the previous section, Algorithm 1 presents how to compute the proper step size $\tau$. First, we use the backtracking strategy to gradually

**Algorithm 1** The algorithm to choose the proper step size for iteration $i$.

```
Input: descent direction d_i, position of rigid body x_i, advancement region (τ_min, τ_max)
Set τ = τ_max
while τ ≥ τ_min do
  x_{i+1} = x_i + τ · d_i
  if U(x_{i+1}) < U(x_i) then
    return τ
  end if
  τ → ρτ {ρ ∈ (0;1) is a decrement factor of the step size. }
end while
return 0
```

reduce $\tau$. In contrast to standard approaches, the initial guess of $\tau$ is neither constant nor depends on the history of the previously accepted step sizes. Instead, it is determined from the advancement region to take into account the magnitude of the generalized force and provide the initial tentative movement of the rigid body. Second, we track only changes in the energy function and not in the generalized force. Albeit the latter is helpful if one wants to determine proximity to a local minimum, we focus on only decreasing the energy because we may hop between several local minima descending in energy without any guarantees on the value of the generalized force. Finally, we stop the line search if there is no appropriate

step size $\tau$ within the advancement region $(\tau_{\min}, \tau_{\max})$. We do not look for the values of $\tau$ smaller than $\tau_{\min}$ since step sizes below this value provide uselessly small movements of the rigid body.

### Algorithm outline

Given the procedure to compute the descent direction, the advancement region, and the algorithm to compute the proper step size, now we present Algorithm 2 for the rigid-body minimization of a molecular complex consisting of $N$ subunits. Here, we iteratively update

**Algorithm 2** The algorithm for the rigid-body minimization of a molecular complex.

**for** $k = 0$ to $K_{\max}$ **do**
  Compute energy function $U_k$
  **for all** molecules $M_i$, $i \in \{1, 2..N\}$ **do**
    Compute descent direction $\mathbf{d}_k^i$
    Compute advancement region $[\tau_{\min}^i, \tau_{\max}^i]$
  **end for**
  Define minimal advancement region $[\hat{\tau}_{\min}, \hat{\tau}_{\max}] = \min_i [\tau_{\min}^i, \tau_{\max}^i]$
  Choose proper step size $\tau_{\text{opt}}^i$ from $[\hat{\tau}_{\min}, \hat{\tau}_{\max}]$ using Algorithm 1 {At this step we find new position of each rigid-body and new value of the energy function $U_{k+1} \leq U_k$}
  **if** $\tau_{\text{opt}}^i = 0 \; \forall i \in \{1, 2..N_{\text{molecules}}\}$ **then**
    **return** {No allowed step sizes are found}
  **end if**
**end for**
**return** {Maximum number of iteration $K_{\max}$ is achieved}

the positions and orientations of subunits in the molecular complex. Given the computed generalized forces and the corresponding advancement regions, we choose the smallest advancement region to guarantee the absence of large movements for any monomer in the molecular complex. The positions are updated only for those monomers where the proper step size are found within the advancement region. If no proper step size within the advancement region is found for any of the monomers, we stop the rigid-body minimization, since smaller step sizes provide negligibly small movements of the monomers. The latter condition is implicitly related with the difference in energy in the two subsequent steps. Indeed, if the stop condition holds, the difference between the energies in the two subsequent steps equals to zero. For this reason we do not use the tolerance criterion for the changes in the energy function and run the algorithm until one can find rigid transformations corresponding to the advancement region of $\tau$ or the maximum number of iteration $K_{\max}$ is achieved.

In Section 5.4 we provide numerical results that demonstrate the power and efficiency of Algorithm 2 when it is used both with a classical force-field and with a knowledge-based scoring function.

## 5.3 Methods

### 5.3.1 Test benchmark for the classical force-field

The first benchmark was generously provided by Dima Kozakov from Boston University. It consists of docking predictions for five different protein docking jobs produced by the ClusPro automated server [28, 71] with the default parameters. For each protein complex, 60 docking predictions were added to the benchmark.

### 5.3.2 Test benchmark for the knowledge-based scoring function

To test our rigid-body minimization approach in combination with the knowledge-based scoring function, we used 844 non-redundant protein-protein complex structures from the database prepared by Huang and Zou [51]. This database contains non-homologous protein-protein complexes extracted from the PDB [13] and includes 655 homodimers and 196 heterodimers. For each native complex, we used the Hex rigid-body docking program [115] to generate docking poses. More precisely, for the Hex input, we used polar Fourier shape expansions to polynomial order $N = 31$, the real-space angular search step of $7.5^\circ$, the radial search range of 40 Å with a translational step of 2.5 Å and the subsequent sub-step of 1.25 Å. We clustered the docking poses with a threshold of 8 Å and left only the docking predictions with non-zero rotational part of the spatial transform. Top 200 docking poses of each native complex were added to the test benchmark, resulting in $200 \times 844 = 169,644$ protein complexes. We compare the docking predictions by assessing the quality of a pose based on the RMSD of the backbone atoms of the ligand ($L_{\text{RMSD}}$) after the receptors in the native and the docking pose conformation have been optimally superimposed (see Table 1.1). We use the fast open-source RigidRMSD library [104] to compute RMSDs given a spatial transform of a docking pose. Finally, we evaluated the success rate of the Hex scoring function on the constructed benchmark according to the value of $L_{\text{RMSD}}$ for comparison.

### 5.3.3 Test benchmark of moderate and large steric clashes

The Hex and Piper energy functions involve penalty terms that prevent large steric clashes to appear in the output predictions. Thus, only soft steric clashes could be present in the

two previous benchmarks. In order to demonstrate that our rigid-body minimization algorithm is able to resolve larger steric clashes, we constructed another two sets of molecular complexes.

The first set consists of five complexes taken from the protein-protein docking benchmark v.4.0 [53], for which the unbound conformations of monomers after superimposition on the corresponding bound conformations possess moderate steric clashes. The latter is assessed by the number of clashed atoms to be in between 30 and 200. Two heavy atoms form a clash if they belong to different monomers and the distance between them is less than 2.4 Å, which is twice the van der Waals radius of a hydrogen atom. Figure 5.2 presents initial conformations of the five complexes (the first column).

For the second set we selected four native complexes from the non-homologous set of protein dimers prepared by Huang and Zou [51] and created large steric clashes (number of clashed atoms is greater than 200) by moving monomers of a protein complex toward each other. Figure 5.3 A presents initial conformations of the four complexes.

## 5.4 Results and Discussion

To demonstrate the power and the advantages of our rigid-body minimization approach we examined it with test cases that have a practical application for computational structural biology. First, we show the relevance of the method for refinement of binding candidates with the CHARMM force-field. To do so, we run CARBON on the test benchmark generated with the Piper docking program [72] and compare the obtained results with the state-of-the-art approach of rigid-body manifold optimization method [92], which was specifically fine-tuned to deal well with Piper docking predictions. Second, on the benchmark generated with the Hex docking program [115], we demonstrate that a combination of the rigid-body minimization algorithm with the knowledge-based potential improves the scoring results. Third, we show that in case of a large steric overlap appearing between subunits of a molecular complex, our algorithm remains stable and resolves the steric clashes properly. Finally, we conclude the section with a general discussion.

### 5.4.1 The CARBON algorithm in combination with classical force-field

For the first test, we use the benchmark produced with the Piper docking program (see Section 5.3.1) and the widely used CHARMM potential as the classical force-field. As the reference method for the comparison, we choose the rigid-body manifold optimization approach (MO) [92]. Their method uses local parametrization of $SO(3) \times R^3$ via the exponential map

Table 5.1 Performance the rigid-body optimization algorithms on the benchmark generated with the Piper docking program. The average difference between the energy values of the final conformations is denoted by av. $\Delta E$. The average $L_{\mathrm{RMSD}}$ between the starting and final conformations is denoted by av. $L_{\mathrm{RMSD}}$. The $L_{\mathrm{RMSD}}$ value is defined as the RMSD of the backbone atoms of the ligand after the receptors in the native and the docking pose conformation have been optimally superimposed. The average number of energy and forces computations is denoted by av. no. of computations. The number of cases where one algorithm was found to be superior to the other in terms of the value of the reached energy and computational efficiency is denoted by no. of wins E and no. of wins N, respectively.

| complex | av. $\Delta E$ (kcal/mol) | av. $L_{\mathrm{RMSD}}$ (Å) | av. no. of computations | no. of wins E | no. of wins N |
|---|---|---|---|---|---|
| | | CARBON/MO | CARBON/MO | CARBON/MO | CARBON/MO |
| 01 | 73.410 | 10.026/10.298 | 87/206 | 0/49 | 42/7 |
| 02 | 41.108 | 11.654/7.194 | 100/174 | 1/46 | 39/8 |
| 03 | 37.649 | 9.006/6.891 | 137/203 | 1/38 | 29/10 |
| 04 | 35,530 | 10.500/6.116 | 98/187 | 1/47 | 41/7 |
| 05 | 25,405 | 14.730/9.081 | 65/210 | 1/40 | 34/7 |
| Total | 42,620 | 11.183/7.916 | 97.4/196 | 1.8%/98.2% | 83.6%/17.4% |

and the limited-memory BFGS minimization algorithm, which is a quasi-Newton method to solve the local minimization problem on a six-dimensional Euclidean space [80]. The parameters of this method were specifically calibrated to work well on docking predictions produced by the Piper software. We implemented the CARBON algorithm inside the C-library source code of MO, provided by Mirzaei et al., such that the computations of energy, forces and neighbor list are the same for both methods. Then, we ran the rigid-body minimization algorithms for each conformation in the benchmark. We discarded the minimized conformations as failures if: *i)* the RMSD between the initial and the final conformations is greater than 30 Å or *ii)* the final conformation contains unresolved steric clashes or *iii)* a method takes more than 500 evaluations of energy and forces. The first criterion assures that the rigid-body minimization does not lead monomers far away from each other. The second criterion discards minimized conformations that still contain steric clashes. The third criterion guarantees that the final conformation is reached sufficiently fast.

To compare performance of the two methods, we measured the average difference between the energy values of the final conformations, the average $L_{\mathrm{RMSD}}$ of the final conformations with respect to the initial conformation, the average number of energy and forces computations, and the number of cases where one algorithm was found to be superior to the other in terms of the value of the reached energy and computational efficiency. Table 5.1 reports the calculated characteristics.

As one can see, the MO method provides final conformations with a lower energy in

almost all the cases in the benchmark. On one hand, this is an expected result since the MO method is specifically fine-tuned to deal well with the Piper docking predictions. On the other hand, we observed that the $L_{\mathrm{RMSD}}$ between the corresponding final conformations produced by the two methods varies in a wide range from 0 to 25 Å and greater than 3 Å in 88% of the cases (not shown in the table). This indicates that the two methods approach two different local minima and it is difficult to rigorously compare the two rigid-body minimization algorithms. Nonetheless, the average difference between the energy values of final conformations is about 40 kcal/mol, which is less than 2.5% of the average final energy. On average, CARBON produces conformations with larger values of $L_{\mathrm{RMSD}}$ with respect to the initial conformations. However, $L_{\mathrm{RMSD}}$ could be controlled with the right bound of the advancement region, which corresponds to the translation of 3 Å in this test. Concerning the computational efficiency, CARBON obtains the final conformations faster than the MO method in more than 80% of cases. We choose the number of energy and force computations as the criterion of computational efficiency since it is the most expensive operation of the minimization algorithms. On average, CARBON was twice faster compared to the MO approach.

We should, however, pay reader's attention to the fact that the stopping criteria for the two methods are very different. While the MO method spends computational time trying to achieve a better value of energy, the CARBON method terminates as soon as the step size gets smaller than the lower bound value $\tau_{min}$. Lowering the tolerance of the MO method, probably, will speed up the calculations, however, it may also result in the different final conformations. Thus, we can only conclude from this test that the CARBON approach is suitable to be used with a classical force-field such as CHARMM and competitive with the state-of-the-art approach.

### 5.4.2 The CARBON algorithm in combination with knowledge-based scoring function

With the growing number of scoring functions aimed to discriminate between near-native and non-native conformations of protein complexes, we believe it is important to develop rigid-body optimization algorithms which refine well the putative binding poses in combination with these functions. Here we choose the KSENIA potential because it is smooth and appropriate for the rigid-body minimization [105]. For the second test, we used the benchmark constructed with the Hex docking program [115]. It consists of rigid-body poses with the assigned quality, which is evaluated according to the value of $L_{\mathrm{RMSD}}$ (see Section 5.3.2). To demonstrate the efficiency of the CARBON method in combination with the

knowledge-based scoring function, we compared the scoring success rates on the initial set of conformations and on the set of minimized conformations. The success rate is defined as the percentage of protein complexes for which docking predictions with quality 1, 2, or 3 are ranked at the top positions. More precisely, first we ranked the docking predictions with respect to the values of KSENIA and computed the success rates for top-one-quality-one, -two or three, top-ten-quality-one or -two, top-one-quality-one and top-ten-quality-one predictions. Then, we optimized each docking pose using a C++ implementation of our rigid-body minimization algorithm (2) and KSENIA as the potential, re-assigned qualities and re-computed the corresponding success rates. Finally, we evaluated the maximum success rates provided by the initial and the optimized docking poses. Figure 5.1 presents the corresponding success rates. From the figure, one may see that the rigid-body minimiza-

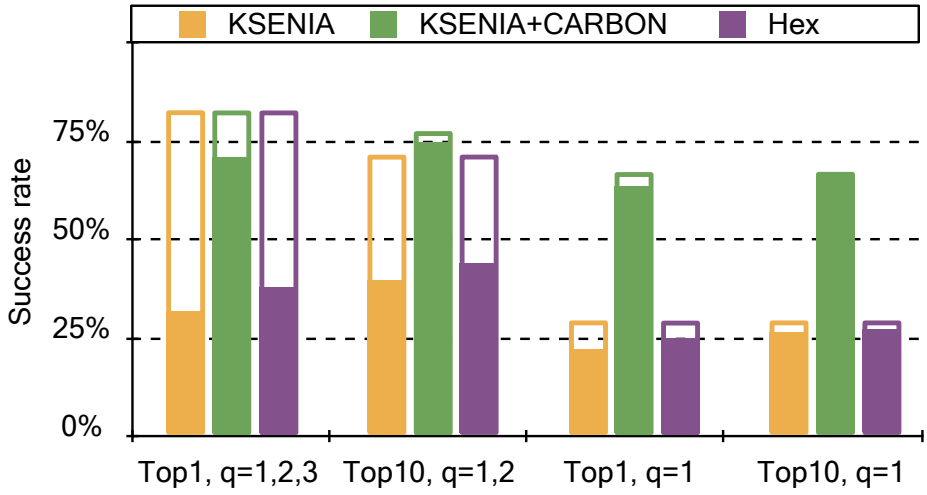


Figure 5.1 Performance of the scoring functions on the test benchmark. Success rates of KSENIA are depicted with the solid yellow rectangles. Success rates of KSENIA along with the rigid-body minimization (KSENIA+CARBON) are depicted with the solid green rectangles. Success rates of the Hex scoring function are depicted with the solid purple rectangles. Hollow rectangles of the corresponding colour represent the maximum achievable success rates. Top$N$ value is defined as the percentage of protein complexes for which at least one of the docking prediction with the corresponding quality $q$ is present within the first $N$ docking poses. The quality of predictions $q$ is evaluated according to the value of $L_{\mathrm{RMSD}}$ (see Table 1.1).

tion dramatically improves the scoring results. In particular, the rigid-body minimization ameliorates poses of quality-two or -three into quality-one. Indeed, the maximum success rate for the top-one-quality-one, -two or -three predictions did not change, while the other maximum success rates increased. Moreover, the total number of quality-one docking poses in the test benchmark was increased by approximately five times (not shown on the figure),

rising the maximum success rate from 28% to 66%. Finally, the improvement provided by the rigid-body minimization makes the corresponding success rates to be more than twice higher compared to both the success rates of Hex and the success rates of scoring without the refinement procedure.

### 5.4.3 The CARBON algorithm to refine moderate and large steric clashes

In the applications described above we use the benchmarks constructed with the rigid-body docking programs. Therein, a docking potential involves a penalty term, which prevents the formation of a large steric clash between the monomers. As a consequence, these benchmarks have no cases with a large overlap between the protein monomers. However, for a general input conformation of a molecular complex, it is important that the rigid-body minimization is also able to resolve large steric clashes. For this reason, in the third test we use the benchmark with moderate and large steric clashes (see Section 5.3.3). For each complex in the benchmark, we performed the rigid-body minimization in combination with the knowledge-based scoring function as well as the classical force-field. In the latter case, we also ran the MO algorithm for the comparison. Table 5.2 lists the energies and the number of clashed atoms before and after the minimization.

Regarding the rigid-body minimization in combination with the CHARMM force-field, for the cases with moderate steric clashes our approach outperforms the MO method in terms of energy and quality of the final conformations in four cases out of five. In all five cases CARBON dramatically improved the conformation of the complexes in terms of its energy as well as resolved almost all presented steric clashes. Despite the enormous values of energy and force of the starting conformations, the monomers in the final conformations are not far away from each other and possess a clear interface of interaction between each other. Figure 5.2 presents the refined conformations with moderate steric clashes from the benchmark. For the large steric clashes, neither the MO method nor CARBON are able to refine the starting conformations with the classical force-field well. We believe that this is due to the rugged shape of the energy landscape provided by the CHARMM force-field for such conformations. In contrast, the CARBON algorithm in combination with KSENIA perfectly resolves moderate and large steric clashes for all the complexes in the benchmark. Again, in all cases, the monomers in their final conformations possess a clear interface of interaction between each other. Figure 5.3 presents the refined conformations with large steric clashes from the benchmark. Furthermore, minimization with the knowledge-based scoring function provides a smaller values of $L_{\mathrm{RMSD}}$ between the initial and the final conformations compared to the minimization with the CHARMM force-field (see Table 5.2). To conclude, our approach outperforms the state-of-the-art MO method with the CHARMM

Table 5.2 Performance of the rigid-body optimization algorithms on the benchmark of moderate and large steric clashes. The energy values of starting and final conformations are denoted by $E_{start}$ and $E_{final}$, respectively. The number of clashed atoms in a starting conformation is denoted by no. of clashes. The number of clashed atoms in a final conformation is denoted by no. of remained clashes. The $L_{RMSD}$ between the starting and the final conformations of the ligands after the receptors is denoted by $L_{RMSD}$. The $L_{RMSD}$ value is defined as the RMSD of the backbone atoms of the ligand after the receptors in the native and the docking pose conformation have been optimally superimposed. Two atoms form a clash if they belong to different subunits and the distance between them is less than 2.4 Å.

| complex | $E_{start}$ (kcal/mol) | | $E_{final}$ (kcal/mol) | | no. | no. of remained clashes | | $L_{RMSD}$ (Å) | |
|---|---|---|---|---|---|---|---|---|---|
| | CHARMM | KSENIA | CHARMM CARBON/MO | KSENIA CARBON | of clashes | CHARMM CARBON/MO | KSENIA CARBON | CHARMM CARBON/MO | KSENIA CARBON |
| Moderate steric clashes | | | | | | | | | |
| 1BKD | $5{,}08\times10^{12}$ | $2{,}67\times10^{3}$ | $-8{,}23\times10^{2}/3{,}81\times10^{8}$ | $-9{,}08\times10^{1}$ | 177 | 0/252 | 2 | 26.6/10.5 | 5.9 |
| 1PXV | $7{,}60\times10^{11}$ | $1{,}11\times10^{3}$ | $-3{,}89\times10^{2}/1{,}52\times10^{8}$ | $-9{,}05\times10^{1}$ | 78 | 2/120 | 0 | 14.7/11.6 | 6.5 |
| 1XQS | $2{,}77\times10^{8}$ | $4{,}53\times10^{2}$ | $-1{,}79\times10^{3}/3{,}42\times10^{4}$ | $-5{,}44\times10^{1}$ | 46 | 0/36 | 2 | 27.9/6.5 | 10.4 |
| 2C0L | $2{,}84\times10^{11}$ | $4{,}36\times10^{2}$ | $1{,}19\times10^{3}/1{,}14\times10^{3}$ | $-8{,}16\times10^{1}$ | 35 | 0/0 | 0 | 6.1/4.6 | 4.8 |
| 2OT3 | $1{,}09\times10^{11}$ | $2{,}29\times10^{3}$ | $-3{,}89\times10^{2}/2{,}25\times10^{7}$ | $-8{,}56\times10^{1}$ | 136 | 0/120 | 0 | 8.6/2.4 | 7.9 |
| Large steric clashes | | | | | | | | | |
| 1A0G | $3{,}61\times10^{12}$ | $1{,}24\times10^{4}$ | $1{,}59\times10^{11}/1{,}31\times10^{10}$ | $-5{,}48\times10^{1}$ | 681 | 708/661 | 0 | 3.6/0.335 | 18.9 |
| 11AS | $2{,}68\times10^{14}$ | $2{,}57\times10^{4}$ | $3{,}59\times10^{11}/1{,}33\times10^{12}$ | $-8{,}79\times10^{1}$ | 1393 | 1503/1716 | 0 | 1.5/7.998 | 24.6 |
| 1A4I | $2{,}89\times10^{16}$ | $2{,}07\times10^{4}$ | $3{,}23\times10^{11}/1{,}15\times10^{12}$ | $-9{,}45\times10^{1}$ | 1139 | 1069/1276 | 0 | 6.1/16.737 | 17.9 |
| 1A7N | $1{,}21\times10^{13}$ | $1{,}50\times10^{4}$ | $7{,}18\times10^{10}/1{,}22\times10^{11}$ | $-1{,}48\times10^{2}$ | 801 | 918/976 | 0 | 5.4/12.790 | 20.1 |

force-field on molecular complexes with moderate steric clashes. The CARBON method in combination with the KSENIA scoring function resolves moderate and large steric clashes efficiently. In general, we believe that rigid-body minimization in combination with a soft knowledge-based scoring function is the method of choice to refine docking predictions.

### 5.4.4 General Discussion

In this section we want to highlight advantages and drawbacks of the proposed method and discuss some important aspects regarding rigid-body optimization of biomolecular complexes. First of all, we want to make readers aware of the possible confusion about the term "local rigid-body minimization". Locality here is considered with respect to the conformation of the complex: the interaction area should not change dramatically upon rigid-body refinement. However, it does not mean that one has to find the closest local minimum of the energy function with respect to rigid transformations. Indeed, typically energy function possess many local minima such that the difference between two conformations corresponding to the neighboring minima could be negligible. Thus, the rigid-body minimization algorithm should take into account the possibility of hopping between several local minima on the energy landscape in order to reach lower energy conformations in the neighborhood of the initial conformation. For these reasons we choose the difference in energy as the acceptance criterion for the rigid-body movement, regardless of the force causing this movement. Another advantage of our algorithm is the advancement region concept. Most of modern rigid-body optimization algorithms employ the standard back-tracking line-search method to find an appropriate step size for a given descent direction. Therein, the initial guess of the step size on the current iteration typically depends on the step size on the previous iteration and the backtracking may continue until irrelevantly small step sizes occur. In contrast, we first determine the advancement region based on the largest and smallest rigid-body movements the user allows. Then, the backtracking line-search is performed within the advancement region. Thus, the step sizes on two subsequent iterations are independent, and each step size corresponds to a relevant rigid-body movement. In case when the line-search does not find any appropriate step size within the advancement region and steric clashes are still present in the final conformation, one may conclude that the energy function is not well suitable for the rigid-body optimization, as it happens, for example, with the CHARMM potential applied to protein conformations with large steric clashes.

Regarding further developments of the proposed method, one may use more sophisticated gradient-based approaches or higher-order optimization techniques in order to speed up the optimization. It would be interesting, for example, to develop and test a hybrid approach, which starts with the rigid-body gradient-based minimization to remove steric

clashes and then switches to a higher-order scheme, for example the one from work of Mirzaei et al [92]. However, we believe that the gradient descent is the method of choice when there is no additional information available about the energy landscape of the starting conformation of a molecular complex.

## 5.5 Conclusion

In this study, CARBON, a novel method for fast rigid-body refinement of molecular complexes is proposed. The rigid-body optimization problem is viewed as the calculation of quasi-static trajectories of rigid bodies influenced by the inverse-inertia-weighted energy gradient. In order to determine the appropriate step size in direction of the net generalized force, the concept of advancement region is introduced. Namely, we compute the advancement region as the interval of step sizes that provide movements of the rigid body within a certain RMSD range from the initial conformation. Then, the standard backtracking line search is applied to find the appropriate step size in this interval. As a result, the CARBON approach guarantees the absence of incorrectly large movements of the rigid-bodies as well as the absence of irrelevantly small movements. We tested and validated CARBON on several benchmarks using both a classical force-field and a knowledge-based scoring function. Particularly, CARBON is suitable to be used with the CHARMM force-field and competitive with the state-of-the-art approach. Using a knowledge-based scoring function we demonstrated that CARBON significantly improves the quality of docking predictions in terms of the $L_{RMSD}$, resulting in higher success rate of the scoring protocol. Finally, we demonstrated that the proposed method remains stable and efficiently resolves moderate and large steric clashes when initial conformations of monomers of a molecular complex overlap. CARBON will be made available as a SAMSON Element for the SAMSON software platform at http://www.samson-connect.net.

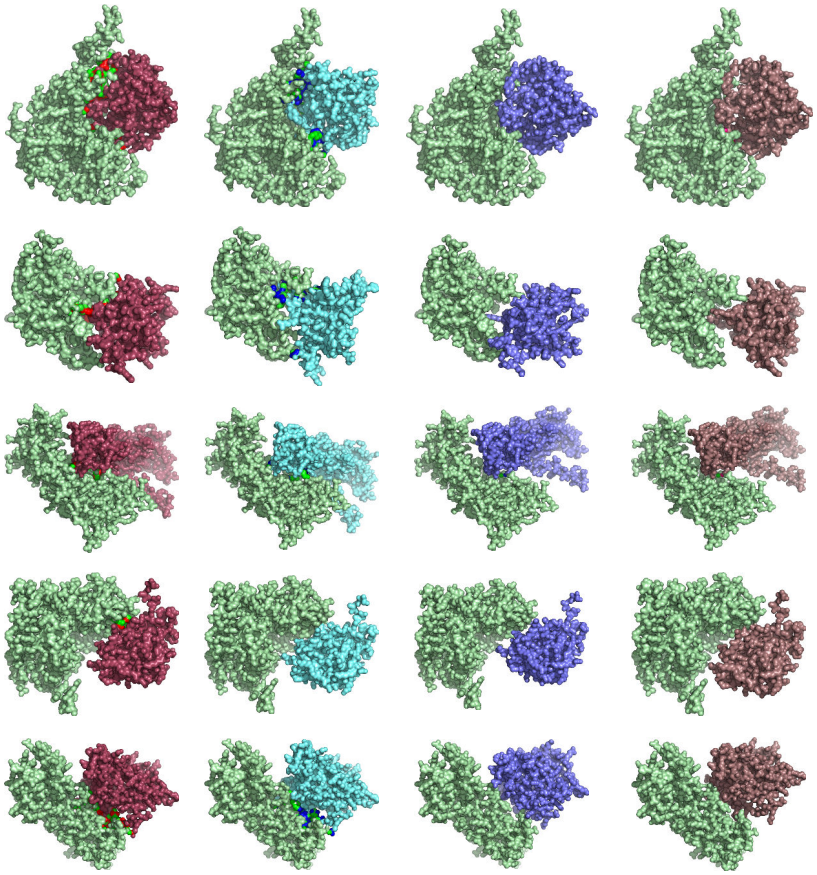

Figure 5.2 Starting and minimized conformations of five complexes: 1BDK, 1PXV, 1XQS, 2C0L, 2OT3, respectively.
**The first column:** starting conformations of the complexes. Receptors and ligands are shown in light green and light red, respectively. The steric clashes are shown in sharp green for the receptors and sharp red for the ligands.
**The second column:** Conformations of the complex after the rigid-body minimization using the MO method and the CHARMM force-field. Receptors and ligands are shown in light green and light blue, respectively. The steric clashes are shown in sharp green for the receptors and sharp blue for the ligands.
**The third column:** Conformations of the complexes after the rigid-body minimization using the CARBON method and the CHARMM force-field. Receptors and ligands are shown in light green and dark blue, respectively. The steric clashes are shown in sharp green for the receptors and sharp blue for the ligands.
**The fourth column:** Conformations of the complexes after the rigid-body minimization using the CARBON method and the KSENIA scoring function. Receptors and ligands are shown in light green and dark orange, respectively. The steric clashes are shown in sharp green for the receptors and sharp magenta for the ligands.
Two heavy atoms form a clash if they belong to different monomers and the distance between them is less than 2.4 Å (twice the van der Waals radius of a hydrogen atom).

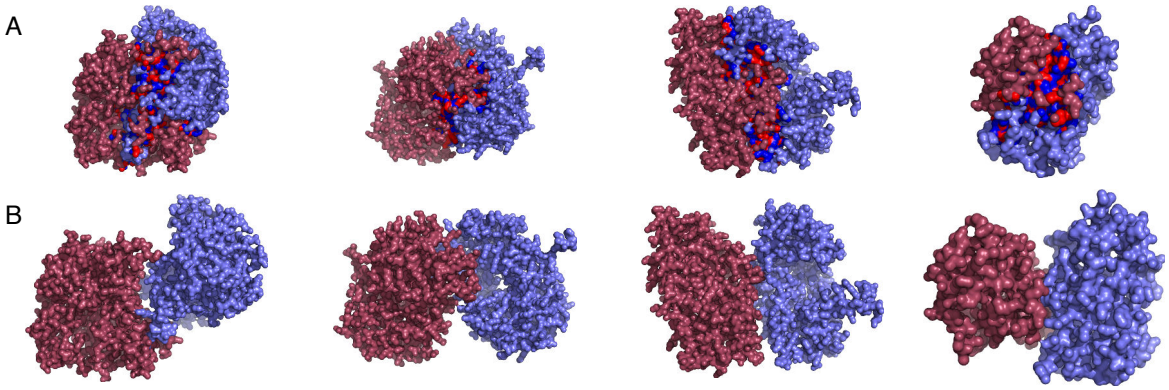


Figure 5.3 Initial and minimized conformations of four complexes: 11AS, 1A0G, 1A4I, 1A7N, respectively. Two monomers are shown in dark red and dark blue, respectively. The clashed atoms of the monomers are colored in sharp red and sharp blue, respectively. Two heavy atoms form a clash if they belong to different monomers and the distance between them is less than 2.4 Å (twice the van der Waals radius of a hydrogen atom).
**A:** Starting conformation of four complexes with a large overlap between the two corresponding monomers. **B:** Final conformation of four complexes with no steric clashes.

# Chapter 6

# A novel criterion to evaluate scoring power of scoring functions for molecular complexes

## 6.1 Introduction

As it is described earlier, direct computation of the binding free energy of proteins is an intractable problem due to its high computational cost, and many various scoring functions are developed to approximate the Gibbs free energy. Thus, it is important to understand the ability of the available scoring functions to distinguish biologically relevant binding candidates among non-relevant ones. From now on we will refer to this ability as to the *scoring power*. A lot of progress was made to assess the scoring power of targeted scoring protocols for virtual screening of protein-ligand complexes [126]. Given libraries of many low-affinity and a few high-affinity compounds, the scoring power of a targeted scoring function is typically measured based on the number of correctly identified active compounds. Different characteristics have been introduced for this assessment, for example, the receiver-operating characteristic (ROC curve) and its variants [66, 132, 139, 141], the enrichment factor (EF) [14, 69, 148], the analysis of variance [125], the area under the accumulation curve (AUAC), the average position of the active compounds [59], the Z-score [34], the robust initial enhancement (RIE) [90, 128], etc.

Whereas some targeted scoring functions significantly enhance the virtual screening performance for particular molecular complexes, there is a growing fundamental and practical interest in developing a general scoring protocol that will well discriminate near-native conformations from the non-native ones, for example, those from protein-protein complexes

[95]. Efficiency of scoring functions is typically assessed using benchmarks that comprise many non-native conformations (decoys) and a few near-native conformations, both obtained with docking algorithms [94]. Therein the scoring power is estimated using the hit rate (HR) criterion. Namely, given a set of native molecular complexes $\{P_i\}$, $i = 1..N$ and a subset of generated conformations $\{P_i^j\}$, $j = 1..N_i$, some of which could be near-native ($\hat{P}_i^j$), the hit rate is introduced as the percentage of near-native complexes in the benchmark ranked at top $M$ positions:

$$HR(M) = \frac{\sum_i^N \mathbb{1}(\exists j < M : P_i^j - \text{near-native})}{N}, \tag{6.1}$$

where $\mathbb{1}$ (condition) is the indicator function, which takes value 1 if the condition holds and 0 otherwise. Near-native conformation can be defined using various similarity metrics, e.g. fraction of native contacts, interface or ligand RMSD, etc [147]. For particular $M$ values of 1, 10 and 100, the HR criterion is also known as Top1, Top10 and Top100 characteristics, respectively. As a consequence of Eq. (6.1), a single scoring function could demonstrate a different scoring power on benchmarks based on the same set of native complexes but with decoys generated with different docking algorithms. Furthermore, the fact that a scoring function can/cannot able to distinguish *one* particular near-native candidate does not imply that it can/cannot distinguish *any* near-native candidate. Thus, the scoring power is a strongly biased criterion, which critically depends on the poses of the binding candidates in the benchmark set.

To address the latter problem, we introduce an alternative criterion to evaluate the scoring power of a scoring function, which is free of the above-mentioned disadvantages. More precisely, we complement the benchmark set with the constructed uniform ensembles of near-native conformations, where each conformation lies within a certain RMSD from the corresponding native conformation. We provide the fast and efficient method to generate the uniform ensembles of near-native conformations. Then, we estimate the scoring power of a scoring function using the cumulative distribution function of decoy scores and the probability density function of the near-native conformation scores. As a result, the obtained characteristic has no bias toward near-native predictions generated by docking algorithms. Thus, the proposed theoretical model could be applied to assess the scoring power of scoring functions for protein-protein as well as for protein-ligand complexes.

To practically demonstrate the proposed criterion, we investigate the scoring power of the pair-wise distance-dependent knowledge-based scoring functions for protein-protein interactions. This class of scoring functions has been recently shown to be very promising compared to the other classes of scoring functions [94]. In this study we derive the

knowledge-based scoring functions using the modern convex optimization apparatus and non-redundant set protein-protein complexes as the training database.

The solution of the convex problem guarantees that the scoring function is optimal, that is, it perfectly distinguishes the native complexes from the non-native ones on the training set. More precisely, the derived scoring function is regularized using the cross-validation technique in order to exclude over-fitting to the training set. Nonetheless, the novel criterion demonstrates that the scoring function discriminates well only those conformations that have ligand-RMSD less than 2 Å, but it loses the scoring power for conformations with ligand-RMSD of 5 Å. Thus, the novel criterion provides a better estimation of the scoring power of scoring functions compared to the standard hit rate criterion. It could be useful in analysis of the scoring power and helpful in better understanding of the properties and pitfalls of different scoring methods. Particularly, the obtained results suggest to look for novel strategies to derive and train knowledge-based scoring functions in order to improve their scoring power.

## 6.2 Theoretical Foundation

### 6.2.1 Near-native Ensemble of Molecular Complex

Generally, the ratio of near-native conformations of molecular complexes produced by the docking algorithms is low. This prevents the rigorous assessment of the success rate of a scoring function on test benchmarks. Here we propose a fast and efficient methodology to construct an ensemble of near-native conformations given the native molecular complex. We start with the equation that relates the axis and the angle of rotation with the RMSD corresponding to the given rotation of a structure [104]:

$$\mathrm{RMSD}^2 = \frac{4}{W}\sin^2\frac{\alpha}{2}\mathbf{n}^T\mathbf{I}\mathbf{n}, \tag{6.2}$$

where $\mathbf{I}$ is the inertia tensor of the structure, $\alpha$ is the angle of rotation about the unit axis $\mathbf{n}$ and $W$ is the sum of atomic weights. Given Eq. (6.2), the rotation angle is expressed as:

$$\alpha = 2\arcsin\left(\frac{\mathrm{RMSD}}{2}\sqrt{\frac{W}{\mathbf{n}^T\mathbf{I}\mathbf{n}}}\right), \tag{6.3}$$

provided that this angle exists. Then, given an axis of rotation $\mathbf{n}$ and a value of RMSD, one can compute rotation angle $\alpha$ that corresponds exactly to RMSD from the initial conformation. Having this, the problem of ensemble generation reduces to collecting of a sufficient

number of rotation axes. For the uniform sampling of the near-native conformations, we collect the rotation axes using the spherical tessellation by an icosahedron. More precisely, starting from an icosahedron with twelve vertices and twenty triangular faces, one connects midpoint of each edge within each face, thus splitting each triangle into four new triangles. Then, this procedure is repeated until a desired level of tessellation is achieved. Finally, the set of normalized radius-vectors to the centroids of each triangle is taken as the collection of the rotation axes. In this study, we use five levels of tessellation resulting in 640 non-collinear rotation axes.

The set of the rotation axes is generated only once. Then, for each native complex in the benchmark and each axis of rotation from the set, one evaluates the rotation angle according to Eq. (6.3) and obtains the corresponding near-native conformation. Thus, the complexity of ensemble generation for a particular native complex is $O(N^l_{\text{atoms}} \times N_{\text{axes}})$, where $N^l_{\text{atoms}}$ is the number of atoms in the ligand and $N_{\text{axes}}$ is the number of rotation axes in the set.

### 6.2.2 Novel Scoring Power Criterion

We start with the introduction of a few concepts useful for the further derivation of the novel success rate equation. First, we will refer to the non-native ensemble as to the non-redundant set of non-native conformations of a particular molecular complex $C$. This ensemble could be generated using various docking algorithms that use exhaustive search in six rotational and translational degrees of freedom. Second, we will refer to the near-native ensemble of complex $C$ corresponding to the RMSD value of $r$ as to the non-redundant set of conformations, such that the RMSD between each conformation and the native one is exactly $r$. Section 6.2.1 describes the efficient algorithm to generate such ensemble. Finally, by assigning a score to each conformation in a particular ensemble, one obtains the score distribution of the ensemble. The concept of score distribution has been already used for targeted scoring functions in virtual screening of protein-ligand molecular complexes by Seifert [125]. However, Seifert used the Gaussian approximation of the score distributions despite the fact that for different targets the distribution varies and could be asymmetric.

Given the score distribution for the near-native ensemble corresponding to the RMSD value of $r$, we reconstruct the probability density function (PDF) $p(x,r)$. The value of $p(x,r)$ at a point $x$ equals to the probability of a random near-native complex with RMSD $= r$ having the score of $x$. We use the kernel density estimation (KDE) function to reconstruct the PDF. In principle, any standard kernel is appropriate. Here we choose the Epanechnikov kernel as the KDE function, because it is optimal in the minimum variance sense.

Given the score distribution for the non-native ensemble, we construct the empirical

cumulative distribution (ECD) function $F(x)$:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\{d_i < x\}, \tag{6.4}$$

where $d_i$ is the score of the $i$th non-native decoy, $N$ is the total number of non-native decoys in the ensemble and $\mathbb{1}\{\text{condition}\}$ is the indicator function. The value of $F(x)$ at a point $x$ corresponds to the probability of the score of a random non-native decoy to be less than $x$. Thus, we can estimate the prediction error of a near-native conformation with the score $s$ as the value $F(s)$ and express the prediction error of a conformation from the near-native ensemble corresponding to the RMSD $= r$ as:

$$e(r) = \int_{-\infty}^{\infty} p(x,r)F(x)dx \tag{6.5}$$

In practice, one is interested in the prediction error of the near-native conformations *below* a certain RMSD value, $r_{\max}$, rather than at fixed RMSD value $r$:

$$E(r_{\max}) = \int_{0}^{r_{\max}} e(r)\rho(r)dr, \tag{6.6}$$

where $\rho(r)$ reflects the probability of a near-native conformation to be at RMSD $= r$ and, hence, depends on the sampling algorithm used to generate the near-native conformations. Here, we obtain the near-native conformations using Eq. 3.14 and the precomputed set of rotation axes (see Section 6.3.1 ). For small angles $\alpha$ the dependence between the RMSD and the angle is linear regardless the axis of rotation:

$$\text{RMSD} \sim \sin\frac{\alpha}{2} \sim \frac{\alpha}{2} \tag{6.7}$$

Thus, using the uniform sampling of $\alpha$ we can treat $\rho(r)$ to be the uniform distribution. Then, the prediction error is written as:

$$E(r_{\max}) = \frac{1}{r_{\max}}\int_{0}^{r_{\max}} e(r)dr \tag{6.8}$$

Recall that $E(r_{\max})$ corresponds to the particular molecular complex $C$. The value of $E(r_{\max}) = 0$ means perfect discrimination between near-native complexes and the decoys, while the value of $E(r_{\max}) = 0.5$ means a random behavior of the scoring function. Thus, for clarity, it is useful to introduce the *correlation coefficient* $R(r_{\max}) = 1 - 2E(r_{\max})$, such that $R(r_{\max}) = 1$ corresponds to the perfect discrimination between near-native complexes and the decoys, $R(r_{\max}) = 0$ for a random behavior of the scoring function, and $R(r_{\max}) = -1$

corresponds to the anti-correlation, that is the near-native complexes possess larger scores compared to the decoys. Finally, given a set of various protein complexes $\{C_i\}$, $i = 1..N$, we evaluate the prediction errors for all the complexes and estimate the scoring power $\Omega$ of the scoring function of interest $\hat{S}$ as:

$$\Omega = \frac{1}{N}\sum_{i=1}^{N} R(r_{\max}, C_i, \hat{S}) \tag{6.9}$$

## 6.3 Materials and Methods

### 6.3.1 Training Set and Test Benchmark

#### Native Complexes

We used the training database of 851 non-redundant protein-protein complex structures prepared by Huang and Zou [51]. This database contains protein-protein complexes extracted from the PDB [12] and includes 655 homodimers and 196 heterodimers. We updated three PDB structures from the original training database: 2Q33 supersedes 1N98, 2ZOY supersedes 1V7B, and 3KKJ supersedes 1YVV. The training database contains only crystal dimeric structures determined by X-ray crystallography at resolution better than 2.5 Å. Each chain of the dimeric structure has at least 10 amino acids, and the number of interacting residue pairs, as defined as having at least 1 heavy atom within 4.5 Å, is at least 30. Each protein-protein interface consists only of 20 standard types of amino acids. No homologous complexes were included in the training database. Two protein complexes were regarded as homologues if the sequence identity between receptor-receptor pairs and between ligand-ligand pairs was $> 70\%$. Finally, Huang and Zou [51] manually inspected the training database and left only those structures that had no artifacts of crystallization.

#### Non-native Decoys

To generate non-native decoys, we used the Hex software for the rigid-body docking [116]. For the Hex input, we used polar Fourier shape expansions to polynomial order $N = 31$, the real-space angular search step of 7.5°, the radial search range of 40 Å with a translational step of 2.5 Å and the subsequent sub-step of 1.25 Å. We ran Hex for each native complex and clustered the docking solutions with a threshold of 8 Å. The first 100 non-native decoys were included in the training set and the first 200 non-native decoys were added for the test benchmark.

### Near-native Decoys

To generate ensembles of near-native decoys we employed the algorithm provided in Section 6.2.1. In this work, we considered five levels of icosahedron tessellation, resulting in 640 non-collinear rotations axes. We used six values of RMSD, namely: 0.5, 1.0, 2.0, 3.0, 4.0 and 5.0 Å. For each native protein complex in the training set we fixed the receptor and generated ensemble of near-native configurations for the ligand. Thus, for each native protein complex we constructed six ensembles corresponding to the different RMSD values, each consisting of $640 \times 2 = 1,280$ near-native complexes (the factor 2 corresponds to the two rotations by angles $\pm\alpha$). Figure 6.1 shows several ligand conformations for the protein complex 1A0G with RMSD value of 5 Å. The near-native complexes along with the Hex decoys form the test benchmark.

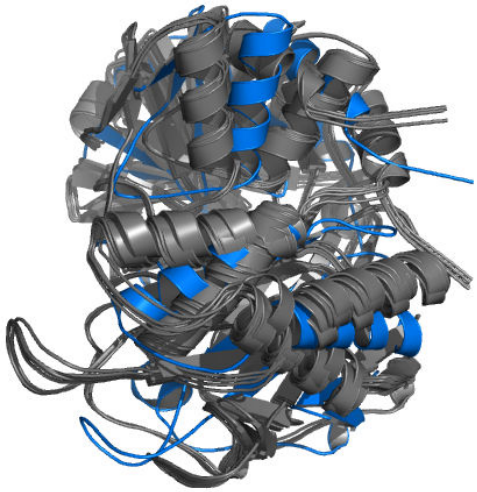

Figure 6.1 Several near-native rigid-body ligand conformations of the protein complex *1A0G*. Each near-native configurations (grey) is exactly 5 Å away from the native configuration (blue).

## 6.3.2 Scoring Function Derivation

To derive pair-wise distance dependent scoring function we used the same concepts of scoring functional $F$ and mapping of molecular structure $P$ to a structure vector $\mathbf{x}$ in a high-dimensional Euclidean space as described in Chapter 4. Given the structure vectors obtained from the training set, in order to determine the scoring vector $\mathbf{w}$, we formulate the convex

optimization problem in the following form:

$$\text{Minimize (in } \mathbf{w}, \mathbf{b}, \text{ )} : \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \sum_{ij}\frac{1}{\gamma}C_{ij}\log\{1+\exp(\gamma y_{ij}(\mathbf{w}^T\mathbf{x}_{ij}+b_i))\}, \tag{6.10}$$

where indexes $i$ runs over different protein complexes, index $j$ runs over the decoys of $i$th protein complex, $C_{ij}$ is the weight of the corresponding structure vector, $y_{ij}$ equals to 1 and $-1$ for the near-native and non-native decoys, respectively, $\mathbf{b}$ is the offset vector, $\gamma$ is the smoothing constant and $\lambda$ is the regularization parameter. There are several differences between the problem (6.10) and the problem (4.8). In the problem (6.10) we use the logarithmic loss function rather than the hinge-loss as was originally introduced in (4.8). This allows to use fast gradient-based methods for minimizing composite objective function to solve the problem (6.10) in its primal form, which turned out to be much faster compared to the block sequential minimal optimization method in the dual form [30]. The regularization parameter $\lambda$ plays a crucial role in the quality of the resulted scoring function and should be optimized using the cross-validation procedure. Here, we optimized parameter $\lambda$ in order to achieve the maximum performance of the scoring function on the benchmark consisting of the Hex decoy structures along with the generated ensembles of near-native structures. We solve the problem (6.10) using the first-order method by Nesterov [99, 100]. The solution of the convex optimization problem, $\mathbf{w}$, is constructed such that near-native conformations possess lower score compared to the non-native conformations for each protein complex in the training set, provided that the corresponding scoring vectors are separable. In other words, the scoring function is built to maximize the performance of the scoring function on the test benchmark in the sense of the HR criterion (Eq. (6.1)).

## 6.4 RESULTS AND DISCUSSION

### 6.4.1 The Score Distribution of Near-native and Non-native Ensembles

Given the scoring function $S$ and the ensemble of decoys structures (either non-native or near-native) one can construct the score distribution by computing the score for each decoy in this ensemble. Figure 6.2 demonstrates such score distributions obtained with the derived knowledge-based scoring function for the near-native ensembles of the protein complex 1A0G corresponding to the RMSD values of 1 Å, 3 Å, and 5 Å, rspectively. As one can see from the figure, the smaller RMSD value is used, the narrower is the score distribution. On average, the standard deviation of the score distribution corresponding to the RMSD value

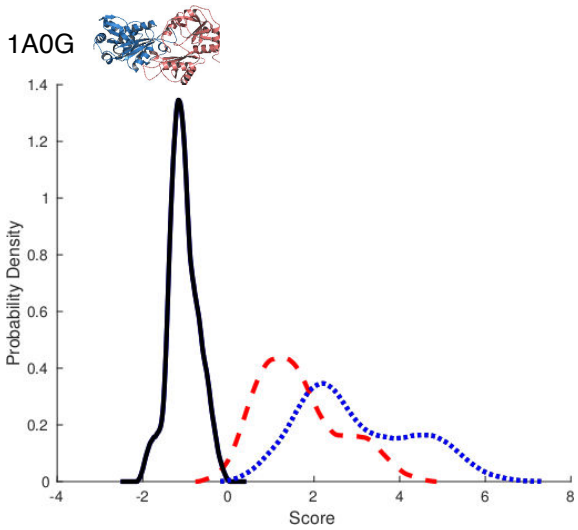


Figure 6.2 The score distributions of the near-native ensembles of protein complex 1A0G corresponding to the RMSD values of 1 Å (black, solid), 3 Å (red, dashed), and 5 Å (blue, dotted).

of 1 Å is 2.3 times smaller compared to the one corresponding to the RMSD value of 5 Å. This is because for small RMSD values the difference between the near-native and the native configurations becomes negligible. Hence, the structure vectors become similar. In the limiting case of the RMSD value approaching 0 Å, they are identically equal. We can also see that for large RMSD values the score distribution is shifted to the right. Indeed, for all protein complexes we observed that the mean value of the score distribution is larger for larger RMSD values. This is because large RMSD values imply a large difference between the decoy and the native conformations. Hence, the scores of the decoys are higher with accordance to the solution of the convex optimization problem (6.10). All these observations are the consequence of the properties of the derived scoring function from the convex optimization problem (6.10).

In a similar manner, one can reconstruct the score distribution of the non-native ensemble. If there is no intersection between the score distributions of the near-native and the non-native ensembles of a particular molecular complex, the scoring function performs perfectly and can thus distinguish any near-native structure from any non-native structure. However, the intersection between the two distributions indicates that the scoring function may fail to discriminate a near-native conformation. Figure 6.3 presents these two cases obtained with the derived scoring function and scoring distributions corresponding to the non-native ensemble and near-native ensembles of RMSD equal to 0.5 Å and 3.0 Å, correspondingly, for protein complex 11AS. In the case when the score distributions of the near-native and the non-native ensembles intersect, one can in principle split the near-native and the non-native

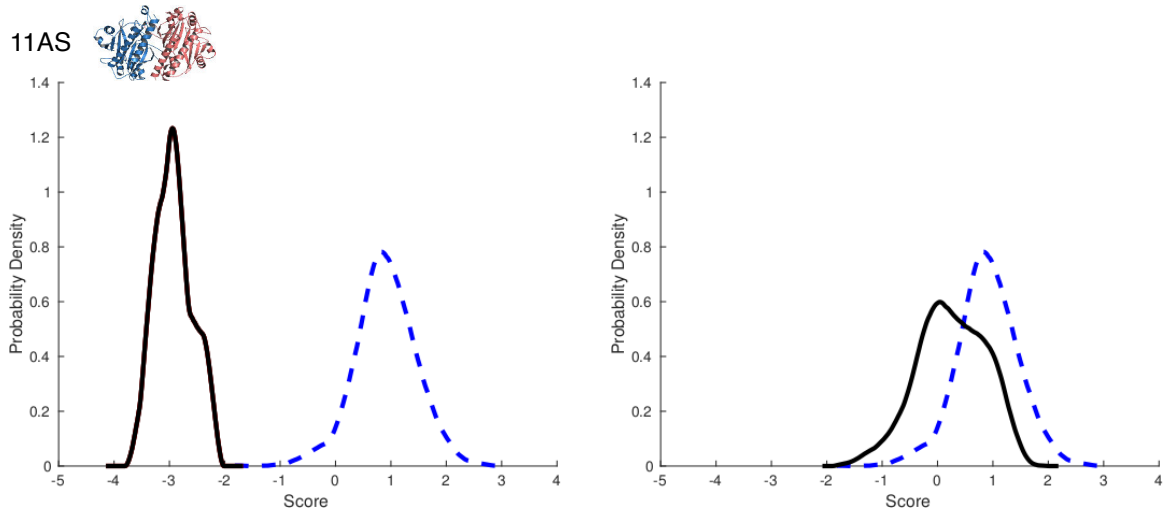


Figure 6.3 **Left:** The near-native score distribution corresponding to the RMSD = 0.5 Å and the non-native decoy distributions (for protein complex 11AS). No intersection between the distributions implies the perfect discrimination between the near-native and the non-native conformations.
**Right:** The near-native score distribution corresponding to the RMSD = 3.0 Å and the non-native decoy distributions (for protein complex 11AS). Intersection indicates that some of the near-native conformations possess a higher score compared to some of the non-native conformations.

ensembles in two parts such that the scoring function performs perfectly on the first ensemble and fails totally on the second one. As a consequence, the standard benchmark-based HR criterion can not correctly evaluate the scoring power of the given scoring function. In contrast, our criterion manipulates with the information about the score distributions instead of a set of scores for a particularly collected benchmark. Thus, it provides a more rigorous assessment of a scoring function.

We want to stress that the score distributions of near-native ensembles depend both on the scoring function and the sampling algorithm, thus, they could be rather different from the normal distribution. In order to demonstrate this fact, we estimated the *goodness of fit* $\tau$ of the reconstructed score distributions $f(x)$ with respect to the Gaussian model of the input set of scores $g(x)$,

$$\tau = \int_{-\infty}^{+\infty} | f(x) - g(x) | dx, \tag{6.11}$$

where $f(x)$ is a score distribution reconstructed with the Epanechnikov kernel and g(x) is the Gaussian distribution corresponding to the mean and the standard deviation values of the set of scores. The $\tau$-characteristic is the area between the two curves and it demonstrates how well the input set of data could be approximated by the Gaussian distribution. Particularly,

$\tau = 0$ corresponds to the normally distributed input data. We calculated the $\tau$-characteristic for the near-native and the non-native ensembles for each protein complex using the derived scoring function and did not find correlation between the $\tau$-characteristics with respect to the RMSD values. Figure 6.4 presents the $\tau$-characteristics for different RMSD values averaged over all the conformations of all protein complexes in the test benchmark and Figure 6.5 presents the obtained $\tau$-characteristics along with the score distributions and the corresponding Gaussian model for two protein complexes (1DDZ and 1PL5).

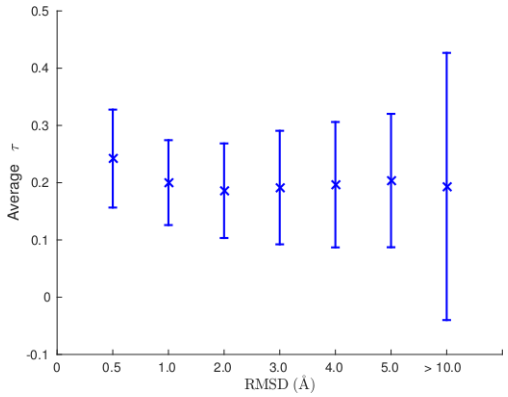


Figure 6.4 Goodness of fit of the reconstructed score distributions with respect to the Gaussian model. Value of the $\tau$-characteristics averaged over the all conformations in the test benchmark are shown with respect to the conformations' RMSD. The error bars corresponds to the standard deviation.

To summarize, we demonstrated that the standard benchmark-based HR criterion could be not rigorous in certain cases and using the score distributions may provide a better estimation of the scoring power of scoring functions. We also demonstrated that the score distributions are different for different protein complexes and, in general, could not be approximated by the normal distribution. Thus, the direct computation of the scores for the near-native and the non-native ensembles for a given scoring function is the inevitable step to reconstruct the corresponding score distributions.

## 6.5 Scoring Power of Pair-wise Distance-dependent Knowledge-based Scoring Function for Protein-protein Interactions

Here, we assess the scoring power of the pair-wise distance-dependent knowledge-based scoring function (SF) derived using the large non-redundant set of 851 protein-protein complexes (see Section 6.3.1) and the modern convex optimization apparatus (see Section 6.3.2). The obtained SF provides Top1 characteristic of 0.92, which means that the scoring function

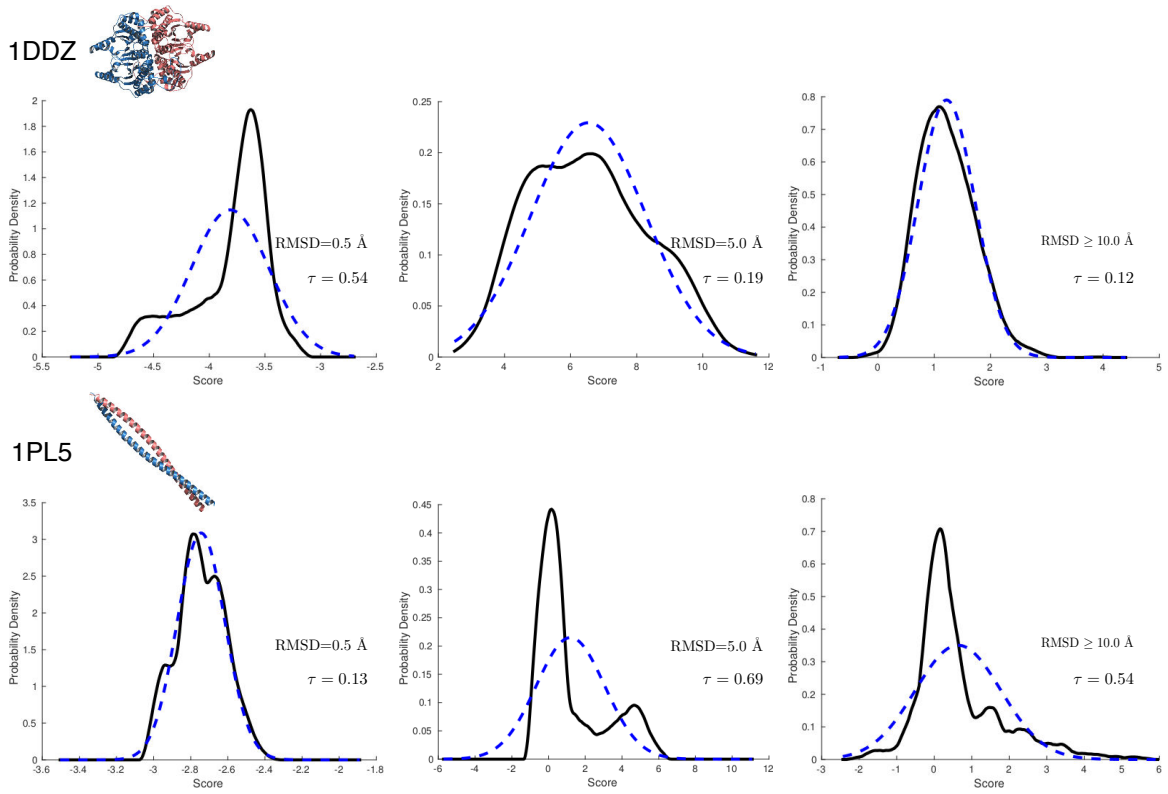


Figure 6.5 Score distributions with the corresponding Gaussian model along with the $\tau$-characteristic obtained for two protein complexes at different RMSD values.

ranks the native complex at the first place in 92 % of protein complexes in the training set. However, using the novel criterion (see Eq. (6.9)), which takes into account the intersection between the score distributions (see Figure 6.3), one can see that the ability to predict near-native structures dramatically decreases as the RMSD between them and the native conformation gets larger (see Table 6.1). As one can see from Table 6.1, for near-native conformations with RMSD less than 0.5 Å the probability to have a higher score for a non-native decoy is less than 1 %. This is the expected result because the SF was trained specifically to well discriminate the native conformations presented in the benchmark. Nonetheless, albeit the derived SF provides good results for near-native conformations within 2 Å from the native structure, it is not able to discriminate near-native conformation of 5 Å (the corresponding scoring power is almost zero). We want to emphasize that since the test benchmark is based on the same native complexes as the training set, the presented $\Omega$ values correspond to the upper estimation of the SF's scoring power.

To conclude, we assessed the scoring power of the pair-wise distance-dependent knowledge-based scoring function for protein-protein interactions using the novel criterion. We demonstrated that albeit the Top1 criterion reports on 92% success rate, the derived SF is able to

Table 6.1 Scoring power of the derived scoring function with respect of the RMSD of near-native conformations.

| RMSD (Å) | Scoring power Ω (see Eq. (6.9)) | Prediction Error $E$ (see Eq. (6.6)) |
|---|---|---|
| 0.5 | 0.982 | 0.008 |
| 1.0 | 0.928 | 0.036 |
| 2.0 | 0.691 | 0.155 |
| 3.0 | 0.419 | 0.291 |
| 4.0 | 0.194 | 0.403 |
| 5.0 | 0.023 | 0.488 |

predict well only near-near native structures within RMSD of 2 Å from the native complex and performs very poorly on the near-native conformation of 5 Å. Thus, the proposed criterion is very useful to analyze the scoring power of a scoring function of interest.

## 6.6 CONCLUSIONS

We propose the novel criterion for a rigorous evaluation of the scoring power of scoring functions. In contrast to the standard hit-rate criterion, which is based on the set of scores computed for the typical benchmarks that comprise few near-native and many non-native conformations, the proposed criterion manipulates with the score distributions of the non-native and the near-native ensembles of conformations. The score distributions depend both on the scoring function and protein complexes and cannot be approximated with the normal distribution. The fast methodology to generate near-native ensembles of conformations with a certain RMSD value is presented. The novel criterion was applied to evaluate the scoring power of the pair-wise distance-dependent knowledge-based scoring function. To derive the scoring function, we used a benchmark consisting of native complexes and many non-native conformations generated with the docking algorithm. The obtained results demonstrate that the scoring function discriminates well near-native conformations with RMSD values within 2 Å. However, it performs poorly for the near-native conformations of higher RMSD values. Thus, the proposed criterion is very useful when doing analysis of a scoring function of interest and should be used instead or at least in combination with the standard hit-rate criterion.

# Chapter 7

# Conclusions

## 7.1 Performance in CAPRI

The algorithms presented in the Thesis were used to predict structures of protein-protein targets in the CAPRI contest [55] as described below. First, we used the Hex software [115] to generate pair-wise docking predictions given the structures of monomers in the unbound state. For the Hex input parameters, we used polar Fourier shape expansions to polynomial order $N = 31$, the real-space angular search step of 7.5°, the radial search range of 40 Å with a translational step of 2.5 Å, the subsequent sub-step of 1.25 Å, and the clustering threshold of $5 - 10$ Å, depending on the number of hits. Generally, we kept about 10,000 docking poses. Then, we used the CARBON rigid-body minimization algorithm in combination with the KSENIA potential to refine the obtained binding candidates. Additionally, the SCWRL4 package[75] was used at each iteration of the rigid-body minimization in order to take into account flexibility of protein side chains. Finally, ten best binding candidates were selected as the submission models for CAPRI.

Figure 7.1 presents the best predictions for protein-protein CAPRI targets on Rounds 26-27 obtained with the described docking pipeline. For the Target 53-54 there were no unbound structure of one of the monomers and thus the homology modeling with I-TASSER server [119] was used in order to generate initial docking models. For Target 53 our docking pipeline succeeded to provide one acceptable-quality prediction among ten top-ranked models. However, there were no successful predictions for Target 54, probably due to the large difference between the true structure and the homologue model (only 4 teams out of 42 succeeded to produce acceptable-quality predictions). For Target 58 we obtained one medium-quality prediction and only four other teams out of 22 succeeded to produce predictions of the same quality.

CAPRI Targets 55 and 56 were aimed to test methods for evaluating the effect of point

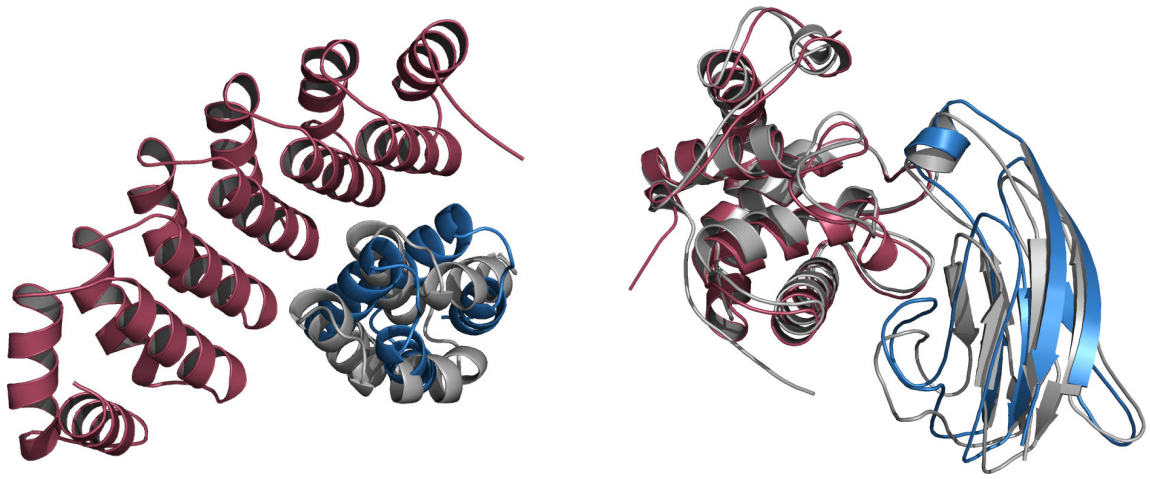

Figure 7.1 The native and predicted structures of the protein-protein complexes for CAPRI Targets.
Left: native structure of Target 53 (grey) and acceptable-quality model produced by the docking pipeline (the two monomers are coloured in red and blue, respectively).
Right: native structure of Target 58 (grey) and medium-quality model produced by the docking pipeline (the two monomers are coloured in red and blue, respectively).

mutations on protein-protein interaction affinity. Predictors were provided with the comprehensive datasets on the effects of every point mutant of two designed protein binders of influenza hemagglutinin [96]. Generally, point mutations stabilized the protein folds and some of them also provided effect on the binding of the complex. It turned out to be very difficult to predict the effect of the mutations following physics-based principles. As a result, only machine-learning methods provided statistically significant correlation between the predicted values and the measured Kd constants. Particularly, we obtained a good correlation between our score and the binding affinity for point mutations corresponding to four residues lying on the interface between the two proteins and failed otherwise [96].

CAPRI Round 30 was launched in collaboration with the Critical Assessment of Structure Predictions of proteins (CASP) [97]. Overall 25 targets were designated as CAPRI comprising 18 protein dimers. In this round we obtained correct predictions for 11 our of 18 dimer targets.

## 7.2 Future Work and General Conclusion

Despite dynamic progress in the computational biology, there are many ways to improve the existing tools and many challenging problems to solve still remain. For example, the DockTrina approach could be generalized for protein oligomers of higher order, where each

pair of monomers interact with each other. The only critical moment for DockTrina is the presence of spatial transforms corresponding to the correct pair-wise conformations in the input file. However, the number of native contacts is decreasing with the growing order of oligomer where all monomers interact with each other, thus, it becomes very difficult to predict correct pair-wise interfaces. Thus, albeit the generalized algorithm is easy to implement, it is hard to test and validate due to the difficulties of composing the corresponding benchmarks. Regarding the improvement of DockTrina, its scoring function (2.2) could be parametrized as follows

$$\text{Score} = \gamma \text{Score}_{\text{Docking}} + \frac{(1-\gamma)\text{Score}^{\max}_{\text{Docking}}}{\alpha \text{RMSD} + \beta}, \tag{7.1}$$

where $\alpha$, $\beta$, and $\gamma$ are parameters to optimize. Then, given collected benchmarks, one can find the optimal values of $\alpha$, $\beta$, and $\gamma$ in order to achieve higher success rates using a particular pair-wise docking program. It would be also interesting to investigate and include other terms in the DockTrina's scoring function, which are not taken into account by scores of a pair-wise docking algorithm, for example a term that reflects the relative interface areas of different pairs of monomers. Finally, it would be interesting to take into consideration some extent of flexibility for the DockTrina predictions.

Concerning development of the knowledge-based scoring function, there are many interesting directions to explore. For example, the results of CAPRI Round 27 suggests that residues located far from the interface do play role in the binding affinity of the protein complex, thus, the corresponding effect should be taken into account in the scoring function derivation. Another point is that the current scoring function is based on the atom typization provided by Huang and Zou [51]. One may consider different atom typization and distinguish atoms not only with respect to their properties, but also, for example, with respect to the type of an amino acid this atom belongs to. Furthermore, the evaluation of derived scoring functions with the novel criterion indicates necessity to take into account more sophisticated statistical information about interactions, e.g. triplet distribution functions, quadruplet distribution functions, etc. Finally, it would be interesting to employ presented methodology in order to develop knowledge-based scoring functions for other types of molecule interactions, such as those involving small ligands, polysaccharides peptides, RNAs and others.

Regarding the rigid-body minimization algorithm, from the obtained results one may see that on one hand, the gradient based CARBON method is very efficient to resolve steric clashes in molecular complexes, and on another hand, the quasi-Newton scheme obtains conformations with a better energy. Thus, it would be interesting to investigate a hybrid

approach that first employs CARBON in order to resolve the steric clashes and then uses a higher order scheme to obtain more reliable energy value. Again, since proteins undergo conformational changes upon binding, it is important to include more degrees of freedom to refine molecular complexes.

Finally, we hope that the developed algorithms will put a weighty bit into the progress in computational biology and will be helpful in solving problems of structure-based drug design in general and protein structure prediction in particular.

# Bibliography


[1] Abbott, A. (2002). Proteomics: the society of proteins. *Nature*, 417(6892):894–896.

[2] André, I., Bradley, P., Wang, C., and Baker, D. (2007). Prediction of the structure of symmetrical protein assemblies. *Proceedings of the National Academy of Sciences*, 104(45):17656–17661.

[3] Andreani, J., Faure, G., and Guerois, R. (2013). Interevscore: a novel coarse-grained interface scoring function using a multi-body statistical potential coupled to evolution. *Bioinformatics*, page btt260.

[4] Andreeva, A., Howorth, D., Chandonia, J.-M., Brenner, S. E., Hubbard, T. J., Chothia, C., and Murzin, A. G. (2008). Data growth and its impact on the scop database: new developments. *Nucleic acids research*, 36(suppl 1):D419–D425.

[5] Bahadur, R. P., Chakrabarti, P., Rodier, F., and Janin, J. (2003). Dissecting subunit interfaces in homodimeric proteins. *Proteins: Structure, Function, and Bioinformatics*, 53(3):708–719.

[6] Bahar, I., Atilgan, A. R., and Erman, B. (1997). Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold Des*, 2(3):173–81.

[7] Baraff, D. (1997). An introduction to physically based modeling: rigid body simulation i - unconstrained rigid body dynamics. *SIGGRAPH Course Notes*.

[8] Beglov, D., Hall, D. R., Brenke, R., Shapovalov, M. V., Dunbrack, R. L., Kozakov, D., and Vajda, S. (2012). Minimal ensembles of side chain conformers for modeling protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 80(2):591–601.

[9] Ben-Naim, A. (1997). Statistical potentials extracted from protein structures: Are these meaningful potentials? *The Journal of Chemical Physics*, 107(9):3698–3706.

[10] Berchanski, A. and Eisenstein, M. (2003). Construction of molecular assemblies via docking: modeling of tetramers with d2 symmetry. *Proteins: Structure, Function, and Bioinformatics*, 53(4):817–829.

[11] Berchanski, A., Segal, D., and Eisenstein, M. (2005). Modeling oligomers with cn or dn symmetry: application to capri target 10. *Proteins: Structure, Function, and Bioinformatics*, 60(2):202–206.

[12] Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., and Jain, S. (2002). The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907.

[13] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.

[14] Bissantz, C., Folkers, G., and Rognan, D. (2000). Protein-based virtual screening of chemical databases. 1. evaluation of different docking/scoring combinations. *Journal of medicinal chemistry*, 43(25):4759–4767.

[15] Bonetta, L. (2010). Interactome under construction. *Nature*, 468(7325):851–854.

[16] Bottema, O. and Roth, B. (1979). *Theoretical Kinematics*. Dover Publ.

[17] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[18] Brent, R. P. and Zimmermann, P. (2011). *Modern computer arithmetic*. Number 18. Cambridge University Press.

[19] Brooks, B. and Karplus, M. (1983). Harmonic dynamics of proteins: Normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proceedings of the National Academy of Sciences*, 80(21):6571–6575.

[20] Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., and Karplus, M. (1983). Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of computational chemistry*, 4(2):187–217.

[21] Cavasotto, C. N., Kovacs, J. A., and Abagyan, R. A. (2005). Representing receptor flexibility in ligand docking through relevant normal modes. *J Am Chem Soc*, 127(26):9632–40.

[22] Chae, M.-H., Krull, F., Lorenzen, S., and Knapp, E.-W. (2010). Predicting protein complex geometries with a neural network. *Proteins: Structure, Function, and Bioinformatics*, 78(4):1026–1039.

[23] Chen, R., Li, L., and Weng, Z. (2003). ZDOCK: an initial-stage protein-docking algorithm. *Proteins Struct. Funct. Bioinf.*, 52(1):80–87.

[24] Chen, R. and Weng, Z. (2002). Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Bioinformatics*, 47(3):281–294.

[25] Chuang, G.-Y., Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2008). Dars (decoys as the reference state) potentials for protein-protein docking. *Biophysical journal*, 95(9):4217–4227.

[26] Chun, H. M., Padilla, C. E., Chin, D. N., Watanabe, M., Karlov, V. I., Alper, H. E., Soosaar, K., Blair, K. B., Becker, O. M., and Caves, L. S. (2000). MBO (n) d: A multibody method for long-time molecular dynamics simulations. *Journal of Computational Chemistry*, 21(3):159–184.

[27] Comeau, S. R. and Camacho, C. J. (2005). Predicting oligomeric assemblies:< i> n</i>-mers a primer. *Journal of structural biology*, 150(3):233–244.

[28] Comeau, S. R., Kozakov, D., Brenke, R., Shen, Y., Beglov, D., and Vajda, S. (2007). Cluspro: performance in capri rounds 6–11 and the new server. *Proteins: Structure, Function, and Bioinformatics*, 69(4):781–785.

[29] Coutsias, E. A., Seok, C., and Dill, K. A. (2004). Using quaternions to calculate RMSD. *J. Comput. Chem.*, 25(15):1849–1857.

[30] Derevyanko, G. and Grudinin, S. (2015). Convex-pp: Predicting protein–protein interactions using polynomial expansions and convex optimisation. unpublished. *to be published*.

[31] Diamond, R. (1988). A note on the rotational superposition problem. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 44(2):211–216.

[32] Esquivel-Rodríguez, J., Yang, Y. D., and Kihara, D. (2012). Multi-lzerd: Multiple protein docking for asymmetric complexes. *Proteins: Structure, Function, and Bioinformatics*, 80(7):1818–1833.

[33] Evans, D. and Murad, S. (1977). Singularity free algorithm for molecular dynamics simulation of rigid polyatomics. *Mol. Phys.*, 34(2):327–331.

[34] Ferrara, P., Gohlke, H., Price, D. J., Klebe, G., and Brooks, C. L. (2004). Assessing scoring functions for protein-ligand interactions. *Journal of medicinal chemistry*, 47(12):3032–3047.

[35] Ferro, D. R. and Hermans, J. (1977). A different best rigid-body molecular fit routine. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 33(2):345–347.

[36] Finkelstein, A., Badretdinov, A., and Gutin, A. (2004). Why do protein architectures have boltzmann-like statistics? *Proteins: Structure, Function, and Bioinformatics*, 23(2):142–150.

[37] Frauenfelder, H., Sligar, S. G., and Wolynes, P. G. (1991). The energy landscapes and motions of proteins. *Science*, 254(5038):1598–1603.

[38] Gavin, A.-C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L. J., Bastuck, S., Dümpelfeld, B., et al. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636.

[39] Golub, G. H. and Loan, C. F. V. (2012). *Matrix Computations*. JHU Press.

[40] Goodsell, D. S. and Olson, A. J. (2000). Structural symmetry and protein function. *Annual review of biophysics and biomolecular structure*, 29(1):105–153.

[41] Gray, J., Moughon, S., Wang, C., Schueler-Furman, O., Kuhlman, B., Rohl, C., and Baker, D. (2003). Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *Journal of Molecular Biology*, 331(1):281–300.

[42] Gwak, S., Kim, J., and Park, F. C. (2003). Numerical optimization on the euclidean group with applications to camera calibration. *Robotics and Automation, IEEE Transactions on*, 19(1):65–74.

[43] Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature*, 402:C47–C52.

[44] Henrick, K. and Thornton, J. M. (1998). Pqs: a protein quaternary structure file server. *Trends in biochemical sciences*, 23(9):358–361.

[45] Hinsen, K. (1998). Analysis of domain motions by approximate normal mode calculations. *Proteins Structure Function and Genetics*, 33(3):417–429.

[46] Hinsen, K. (2000). The molecular modeling toolkit: a new approach to molecular simulations. *Journal of Computational Chemistry*, 21(2):79–85.

[47] Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, 4(4):629–642.

[48] Horn, B. K., Hilden, H. M., and Negahdaripour, S. (1988). Closed-form solution of absolute orientation using orthonormal matrices. *J. Opt. Soc. Am. A.*, 5(7):1127–1135.

[49] Huang, P.-S., Love, J. J., and Mayo, S. L. (2005). Adaptation of a fast fourier transform-based docking algorithm for protein design. *Journal of computational chemistry*, 26(12):1222–1232.

[50] Huang, S.-Y., Yan, C., Grinter, S. Z., Chang, S., Jiang, L., and Zou, X. (2013). Inclusion of the orientational entropic effect and low-resolution experimental information for protein–protein docking in critical assessment of PRedicted interactions (CAPRI). *Proteins: Structure, Function, and Bioinformatics*, 81(12):2183–2191.

[51] Huang, S.-Y. and Zou, X. (2008). An iterative knowledge-based scoring function for protein–protein recognition. *Proteins: Structure, Function, and Bioinformatics*, 72(2):557–579.

[52] Hwang, H., Pierce, B., Mintseris, J., Janin, J., and Weng, Z. (2008). Protein–protein docking benchmark version 3.0. *Proteins: Structure, Function, and Bioinformatics*, 73(3):705–709.

[53] Hwang, H., Vreven, T., Janin, J., and Weng, Z. (2010). Protein–protein docking benchmark version 4.0. *Proteins: Structure, Function, and Bioinformatics*, 78(15):3111–3114.

[54] Inbar, Y., Benyamini, H., Nussinov, R., and Wolfson, H. J. (2005). Prediction of multimolecular assemblies by multiple docking. *Journal of molecular biology*, 349(2):435–447.

[55] Janin, J. (2005). Assessing predictions of protein–protein interaction: the capri experiment. *Protein science*, 14(2):278–283.

[56] Janin, J., Henrick, K., Moult, J., Eyck, L. T., Sternberg, M. J., Vajda, S., Vakser, I., and Wodak, S. J. (2003). CAPRI: a critical assessment of predicted interactions. *Proteins: Structure, Function, and Bioinformatics*, 52(1):2–9.

[57] Jorgensen, W. L., Maxwell, D. S., and Tirado-Rives, J. (1996). Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236.

[58] Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 32(5):922–923.

[59] Kairys, V., Fernandes, M. X., and Gilson, M. K. (2006). Screening drug-like compounds by docking to homology models: a systematic study. *Journal of chemical information and modeling*, 46(1):365–379.

[60] Karaca, E., Melquiond, A. S., de Vries, S. J., Kastritis, P. L., and Bonvin, A. M. (2010). Building macromolecular assemblies by information-driven docking introducing the haddock multibody docking server. *Molecular & Cellular Proteomics*, 9(8):1784–1794.

[61] Karney, C. F. (2007). Quaternions in molecular modeling. *J. Mol. Graphics Modell.*, 25(5):595–604.

[62] Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., and Vakser, I. A. (1992). Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl. Acad. Sci. U.S.A.*, 89(6):2195.

[63] Kearsley, S. K. (1989). On the orthogonal transformation used for structural comparisons. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 45(2):208–210.

[64] Kim, Y. C. and Hummer, G. (2008). Coarse-grained models for simulations of multiprotein complexes: application to ubiquitin binding. *Journal of molecular biology*, 375(5):1416–1433.

[65] Kirys, T., Ruvinsky, A. M., Tuzikov, A. V., and Vakser, I. A. (2012). Rotamer libraries and probabilities of transition between rotamers for the side chains in protein–protein binding. *Proteins: Structure, Function, and Bioinformatics*, 80(8):2089–2098.

[66] Klon, A. E., Glick, M., and Davies, J. W. (2004). Combination of a naive bayes classifier with consensus scoring improves enrichment of high-throughput docking results. *Journal of medicinal chemistry*, 47(18):4356–4359.

[67] Kneller, G. R. (1991). Superposition of molecular structures using quaternions. *Mol. Simul.*, 7(1-2):113–119.

[68] Kobe, B., Guncar, G., Buchholz, R., Huber, T., Maco, B., Cowieson, N., Martin, J. L., Marfori, M., and Forwood, J. K. (2008). Crystallography and protein-protein interactions: biological interfaces and crystal contacts. *Biochemical Society transactions*, 36(Pt 6):1438–1441.

[69] Kontoyianni, M., McClellan, L. M., and Sokol, G. S. (2004). Evaluation of docking performance: comparative data on docking algorithms. *Journal of medicinal chemistry*, 47(3):558–565.

[70] Koppensteiner, W. and Sippl, M. (1998). Knowledge-based potentials–back to the roots.

[71] Kozakov, D., Beglov, D., Bohnuud, T., Mottarella, S. E., Xia, B., Hall, D. R., and Vajda, S. (2013). How good is automated protein docking? *Proteins: Structure, Function, and Bioinformatics*, 81(12):2159–2166.

[72] Kozakov, D., Brenke, R., Comeau, S. R., and Vajda, S. (2006). Piper: An fft-based protein docking program with pairwise potentials. *Proteins: Structure, Function, and Bioinformatics*, 65(2):392–406.

[73] Krissinel, E. (2010). Crystal contacts as nature's docking solutions. *Journal of computational chemistry*, 31(1):133–143.

[74] Krissinel, E. and Henrick, K. (2007). Inference of macromolecular assemblies from crystalline state. *Journal of molecular biology*, 372(3):774–797.

[75] Krivov, G. G., Shapovalov, M. V., and Dunbrack, R. L. (2009). Improved prediction of protein side-chain conformations with scwrl4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795.

[76] Leelananda, S. P., Feng, Y., Gniewek, P., Kloczkowski, A., and Jernigan, R. L. (2011). Statistical contact potentials in protein coarse-grained modeling: from pair to multi-body potentials. In *Multiscale Approaches to Protein Modeling*, pages 127–157. Springer.

[77] Lesk, A. M. (1986). A toolkit for computational molecular biology. II. on the optimal superposition of two sets of coordinates. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 42(2):110–113.

[78] Levy, E. D., Pereira-Leal, J. B., Chothia, C., and Teichmann, S. A. (2006). 3d complex: a structural classification of protein complexes. *PLoS computational biology*, 2(11):e155.

[79] Li, Z., Gou, J., and Chu, Y. (1998). Geometric algorithms for workpiece localization. *Robotics and Automation, IEEE Transactions on*, 14(6):864–878.

[80] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

[81] Liu, P., Agrafiotis, D. K., and Theobald, D. L. (2010). Fast determination of the optimal rotational matrix for macromolecular superpositions. *J. Comput. Chem.*, 31(7):1561–1563.

[82] Lu, H. and Skolnick, J. (2001). A distance-dependent atomic knowledge-based potential for improved protein structure selection. *Proteins: Structure, Function, and Bioinformatics*, 44(3):223–232.

[83] Magrane, M., Consortium, U., et al. (2011). Uniprot knowledgebase: a hub of integrated protein data. *Database*, 2011:bar009.

[84] Maiorov, V. N. and Grippen, G. M. (1992). Contact potential that recognizes the correct folding of globular proteins. *Journal of molecular biology*, 227(3):876–888.

[85] Mandell, J. G., Roberts, V. A., Pique, M. E., Kotlovyi, V., Mitchell, J. C., Nelson, E., Tsigelny, I., and Ten Eyck, L. F. (2001). Protein docking using continuum electrostatics and geometric fit. *Protein Eng.*, 14(2):105–113.

[86] Mascioni, A., Karim, C., Zamoon, J., Thomas, D. D., and Veglia, G. (2002). Solid-state nmr and rigid body molecular dynamics to determine domain orientations of monomeric phospholamban. *J. Am. Chem. Soc.*, 124(32):9392–3.

[87] Mashiach, E., Schneidman-Duhovny, D., Andrusier, N., Nussinov, R., and Wolfson, H. J. (2008). Firedock: a web server for fast interaction refinement in molecular docking. *Nucleic acids research*, 36(suppl 2):W229–W232.

[88] Mashiach-Farkash, E., Nussinov, R., and Wolfson, H. J. (2011). Symmref: A flexible refinement method for symmetric multimers. *Proteins: Structure, Function, and Bioinformatics*, 79(9):2607–2623.

[89] May, A. and Zacharias, M. (2008). Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein-protein docking. *Proteins*, 70(3):794–809.

[90] McGaughey, G. B., Sheridan, R. P., Bayly, C. I., Culberson, J. C., Kreatsoulas, C., Lindsley, S., Maiorov, V., Truchon, J.-F., and Cornell, W. D. (2007). Comparison of topological, shape, and docking methods in virtual screening. *Journal of chemical information and modeling*, 47(4):1504–1519.

[91] McLachlan, A. D. (1982). Rapid comparison of protein structures. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 38(6):871–873.

[92] Mirzaei, H., Beglov, D., Paschalidis, I. C., Vajda, S., Vakili, P., and Kozakov, D. (2012). Rigid body energy minimization on manifolds for molecular docking. *Journal of chemical theory and computation*, 8(11):4374–4380.

[93] Miyazawa, S. and Jernigan, R. (1985). Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552.

[94] Moal, I. H., Moretti, R., Baker, D., and Fernández-Recio, J. (2013a). Scoring functions for protein–protein interactions. *Current opinion in structural biology*, 23(6):862–867.

[95] Moal, I. H., Torchala, M., Bates, P. A., and Fernández-Recio, J. (2013b). The scoring of poses in protein-protein docking: current capabilities and future directions. *BMC bioinformatics*, 14(1):286.

[96] Moretti, R., Fleishman, S. J., Agius, R., Torchala, M., Bates, P. A., Kastritis, P. L., Rodrigues, J. P., Trellet, M., Bonvin, A. M., Cui, M., et al. (2013). Community-wide evaluation of methods for predicting the effect of mutations on protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 81(11):1980–1987.

[97] Moult, J., Fidelis, K., Kryshtafovych, A., Schwede, T., and Tramontano, A. (2014). Critical assessment of methods of protein structure prediction (casp)-round x. *Proteins: Structure, Function, and Bioinformatics*, 82(52):1–6.

[98] Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer journal*, 7(4):308–313.

[99] Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161.

[100] Nesterov, Y. and Nemirovski, A. (2013). On first-order algorithms for l 1/nuclear norm minimization. *Acta Numerica*, 22:509–575.

[101] Pereira-Leal, J. B., Levy, E. D., and Teichmann, S. A. (2006). The origins and evolution of functional modules: lessons from protein complexes. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 361(1467):507–517.

[102] Pierce, B., Tong, W., and Weng, Z. (2005). M-zdock: a grid-based approach for cn symmetric multimer docking. *Bioinformatics*, 21(8):1472–1478.

[103] Pierce, B. G., Hourai, Y., and Weng, Z. (2011). Accelerating protein docking in ZDOCK using an advanced 3D convolution library. *PloS one*, 6(9):e24657.

[104] Popov, P. and Grudinin, S. (2014). Rapid determination of rmsds corresponding to macromolecular rigid body motions. *Journal of computational chemistry*, 35(12):950–956.

[105] Popov, P. and Grudinin, S. (2015). Knowledge of native protein-protein interfaces is sufficient to construct predictive models for the selection of binding candidates. *to be submitted*.

[106] Popov, P., Iouditsky, A., and Grudinin, S. (2015a). On evaluation of scoring functions for molecular interactions. *to be submitted*.

[107] Popov, P., Redon, S., and Grudinin, S. (2015b). Carbon: Controlled-avancement rigid-body optimization of nanosystems. *to be submitted*.

[108] Popov, P., Ritchie, D. W., and Grudinin, S. (2014). Docktrina: Docking triangular protein trimers. *Proteins Struct. Funct. Bioinf.*, 82:34–44.

[109] Powell, M. J. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

[110] Qiu, J. and Elber, R. (2005). Atomically detailed potentials to recognize native and approximate protein structures. *Proteins: Structure, Function, and Bioinformatics*, 61(1):44–55.

[111] Rajgaria, R., McAllister, S., and Floudas, C. (2006). A novel high resolution c$\alpha$ - c$\alpha$ distance dependent force field based on a high quality decoy set. *Proteins: Structure, Function, and Bioinformatics*, 65(3):726–741.

[112] Rarey, M., Wefing, S., and Lengauer, T. (1996). Placement of medium-sized molecular fragments into active sites of proteins. *J. Comput.-Aided Mol. Des.*, 10(1):41–54.

[113] Ravikant, D. and Elber, R. (2010). Pie—efficient filters and coarse grained potentials for unbound protein–protein docking. *Proteins: Structure, Function, and Bioinformatics*, 78(2):400–419.

[114] Ritchie, D. W. and Kemp, G. J. (2000). Protein docking using spherical polar fourier correlations. *Proteins Struct. Funct. Bioinf.*, 39(2):178–194.

[115] Ritchie, D. W., Kozakov, D., and Vajda, S. (2008). Accelerating and focusing protein–protein docking correlations using multi-dimensional rotational fft generating functions. *Bioinformatics*, 24(17):1865–1873.

[116] Ritchie, D. W. and Venkatraman, V. (2010). Ultra-fast fft protein docking on graphics processors. *Bioinformatics*, 26(19):2398–2405.

[117] Rodrigues, J. P. G. L. M., Trellet, M., Schmitz, C., Kastritis, P., Karaca, E., Melquiond, A. S. J., and Bonvin, A. M. J. J. (2012). Clustering biomolecular complexes by residue contacts similarity. *Proteins Struct. Funct. Bioinf.*, 80(7):1810–1817.

[118] Rohl, C. A., Strauss, C. E., Misura, K. M., and Baker, D. (2004). Protein structure prediction using rosetta. *Methods in enzymology*, 383:66–93.

[119] Roy, A., Kucukural, A., and Zhang, Y. (2010). I-tasser: a unified platform for automated protein structure and function prediction. *Nature protocols*, 5(4):725–738.

[120] Saladin, A., Fiorucci, S., Poulain, P., Prévost, C., and Zacharias, M. (2009). Ptools: an opensource molecular docking library. *BMC structural biology*, 9(1):27.

[121] Samudrala, R. and Moult, J. (1998). An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. *Journal of molecular biology*, 275(5):895–916.

[122] Schneidman-Duhovny, D., Hammel, M., and Sali, A. (2011). Macromolecular docking restrained by a small angle x-ray scattering profile. *Journal of structural biology*, 173(3):461–471.

[123] Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., and Wolfson, H. J. (2005). Geometry-based flexible and symmetric protein docking. *Proteins: Structure, Function, and Bioinformatics*, 60(2):224–231.

[124] Schrödinger, L. (2010). The pymol molecular graphics system, version 1.3 r1. *PyMOL, The PyMOL Molecular Graphics System, Version*, 1.

[125] Seifert, M. H. (2006). Assessing the discriminatory power of scoring functions for virtual screening. *Journal of chemical information and modeling*, 46(3):1456–1465.

[126] Seifert, M. H. (2009). Targeted scoring functions for virtual screening. *Drug discovery today*, 14(11):562–569.

[127] Shen, M.-y. and Sali, A. (2006). Statistical potential for assessment and prediction of protein structures. *Protein science*, 15(11):2507–2524.

[128] Sheridan, R. P., Singh, S. B., Fluder, E. M., and Kearsley, S. K. (2001). Protocols for bridging the peptide to nonpeptide gap in topological similarity searches. *Journal of chemical information and computer sciences*, 41(5):1395–1406.

[129] Sillitoe, I., Cuff, A. L., Dessailly, B. H., Dawson, N. L., Furnham, N., Lee, D., Lees, J. G., Lewis, T. E., Studer, R. A., Rentzsch, R., et al. (2012). New functional families (funfams) in cath to improve the mapping of conserved functional sites to 3d structures. *Nucleic acids research*, page gks1211.

[130] Sippl, M. (1990). Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of Molecular Biology*, 213(4):859–883.

[131] Sippl, M., Ortner, M., Jaritz, M., Lackner, P., and Flöckner, H. (1996). Helmholtz free energies of atom pair interactions in proteins. *Folding and Design*, 1(4):289–298.

[132] Swamidass, S. J., Azencott, C.-A., Daily, K., and Baldi, P. (2010). A croc stronger than roc: measuring, visualizing and optimizing early retrieval. *Bioinformatics*, 26(10):1348–1356.

[133] Tama, F., Gadea, F. X., Marques, O., and Sanejouand, Y. H. (2000). Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins*, 41(1):1–7.

[134] Tanaka, S. and Scheraga, H. A. (1976). Medium- and Long-Range Interaction Parameters between Amino Acids for Predicting Three-Dimensional Structures of Proteins. *Macromolecules*, 9(6):945–950.

[135] Theobald, D. L. (2005). Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, 61(4):478–480.

[136] Thomas, P. and Dill, K. (1996). Statistical potentials extracted from protein structures: how accurate are they? *Journal of Molecular Biology*, 257(2):457–469.

[137] Tirion, M. M. (1996). Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Physical review letters*, 77(9):1905.

[138] Tobi, D. and Bahar, I. (2006). Optimal design of protein docking potentials: efficiency and limitations. *Proteins: Structure, Function, and Bioinformatics*, 62(4):970–981.

[139] Triballeau, N., Acher, F., Brabet, I., Pin, J.-P., and Bertrand, H.-O. (2005). Virtual screening workflow development guided by the "receiver operating characteristic" curve approach. application to high-throughput docking on metabotropic glutamate receptor subtype 4. *Journal of medicinal chemistry*, 48(7):2534–2547.

[140] Tron, R. and Vidal, R. (2009). Distributed image-based 3-d localization of camera sensor networks. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, page 901–908. IEEE.

[141] Truchon, J.-F. and Bayly, C. I. (2007). Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem. *Journal of chemical information and modeling*, 47(2):488–508.

[142] Tunbridge, I., Best, R., Gain, J., and Kuttel, M. (2010). Simulation of coarse-grained protein- protein interactions with graphics processing units. *J. Chem. Theory Comput.*

[143] Vajda, S., Hall, D. R., and Kozakov, D. (2013). Sampling and scoring: A marriage made in heaven. *Proteins: Structure, Function, and Bioinformatics*, 81(11):1874–1884.

[144] Vakser, I. A. (2014). Protein-protein docking: From interaction to interactome. *Biophysical Journal*, 107(8):1785–1793.

[145] Venkatraman, V. and Ritchie, D. W. (2012). Flexible protein docking refinement using pose-dependent normal mode analysis. *Proteins: Structure, Function, and Bioinformatics*, 80(9):2262–2274.

[146] Vreven, T., Hwang, H., and Weng, Z. (2013). Exploring angular distance in protein-protein docking algorithms. *PLoS One*, 8(2):e56645.

[147] Wallin, S., Farwer, J., and Bastolla, U. (2003). Testing similarity measures with continuous and discrete protein models. *Proteins Struct. Funct. Bioinf.*, 50(1):144–157.

[148] Warren, G. L., Andrews, C. W., Capelli, A.-M., Clarke, B., LaLonde, J., Lambert, M. H., Lindvall, M., Nevins, N., Semus, S. F., Senger, S., et al. (2006). A critical assessment of docking programs and scoring functions. *Journal of medicinal chemistry*, 49(20):5912–5931.

[149] Wilson, E. B. (1955). *Molecular vibrations: the theory of infrared and Raman vibrational spectra*. Courier Dover Publications.

[150] Zacharias, M. (2003). Protein–protein docking with a reduced protein model accounting for side-chain flexibility. *Protein Science*, 12(6):1271–1282.

[151] Zacharias, M. (2010). Accounting for conformational changes during protein–protein docking. *Current opinion in structural biology*, 20(2):180–186.

[152] Zhang, C., Liu, S., Zhu, Q., and Zhou, Y. (2005). A knowledge-based energy function for protein-ligand, protein-protein, and protein-DNA complexes. *Journal of medicinal chemistry*, 48(7):2325–2335.

[153] Zhou, H. and Zhou, Y. (2002). Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein science*, 11(11):2714–2726.